



UNIVERSIDAD DE LA RIOJA

TRABAJO FIN DE ESTUDIOS

Título

Metodologías híbridas de gestión de proyectos. Propuesta de implantación para la gestión de proyectos software con equipos de trabajo distribuidos

Autor/es

LAURA ÁLVAREZ CRISTÓBAL

Director/es

FRANCISCO JAVIER MARTÍNEZ DE PISÓN ASCACÍBAR

Facultad

Escuela de Máster y Doctorado de la Universidad de La Rioja

Titulación

Máster Universitario en Dirección de Proyectos

Departamento

INGENIERÍA MECÁNICA

Curso académico

2018-19



Metodologías híbridas de gestión de proyectos. Propuesta de implantación para la gestión de proyectos software con equipos de trabajo distribuidos, de LAURA ÁLVAREZ CRISTÓBAL

(publicada por la Universidad de La Rioja) se difunde bajo una Licencia Creative Commons Reconocimiento-NoComercial-SinObraDerivada 3.0 Unported. Permisos que vayan más allá de lo cubierto por esta licencia pueden solicitarse a los titulares del copyright.

Trabajo de Fin de Máster

**Metodologías híbridas de
gestión de proyectos.
Propuesta de implantación para
la gestión de proyectos
software con equipos de
trabajo distribuidos.**

Autora

Laura Álvarez Cristóbal

Tutor: Francisco Javier Martínez de Pisón Ascacíbar

MÁSTER:

Máster en Dirección de Proyectos (851M)

Escuela de Máster y Doctorado



**UNIVERSIDAD
DE LA RIOJA**

AÑO ACADÉMICO: 2018/2019

1. INTRODUCCIÓN Y PLANTEAMIENTO DEL PROBLEMA	1
1.1. Origen de las metodologías Ágiles en la gestión de proyectos	1
1.2. Dificultades de la aplicación de <i>Agile</i> en proyectos a gran escala.....	3
1.3. Metodologías híbridas de gestión de proyectos	5
2. OBJETIVOS Y PLAN DE TRABAJO	7
3. ESTADO DEL ARTE	9
3.1. Gestión de proyectos: antecedentes	9
3.1.1. <i>Gestión de proyectos tradicional</i>	9
3.1.2. <i>Gestión de proyectos Agile</i>	11
3.1.3. <i>Ventajas, inconvenientes y aplicaciones de cada enfoque</i>	15
3.2. Escalado <i>Agile</i>	17
3.2.1. <i>Metodologías Híbridas de gestión de proyectos</i>	17
3.2.2. <i>Experiencias de hibridación</i>	27
3.2.3. <i>Limitaciones en los estudios y conclusiones</i>	31
3.3. Scaled Agile Framework (SAFe)	33
3.3.1. <i>Mentalidad y principios SAFe</i>	33
3.3.2. <i>Configuraciones SAFe</i>	34
3.4. Técnicas y herramientas <i>Agile</i>	40
3.5. Métricas <i>Agile</i>	43
4. PROPUESTA DE METODOLOGÍA HÍBRIDA	47
4.1. Objetivos y justificación de la metodología.....	47
4.2. Selección de frameworks para la gestión del portfolio.....	48
4.3. Hibridación PMI-PMBok / SAFe	50
4.4. Procedimiento de gestión de proyectos.....	57
4.5. Implantación del procedimiento	62
5. CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS	65
6. BIBLIOGRAFÍA	69

Resumen

Las metodologías ágiles (*Agile*) de gestión de proyectos han supuesto una transformación dentro del campo de desarrollo software. La rápida adaptación al cambio y la estrecha colaboración con el cliente son dos de sus pilares básicos. Sin embargo, tras unos años de experiencias en el uso de *Agile*, se ha comprobado que su implantación bajo ciertas circunstancias, no resulta eficiente. Tal es el caso de la aplicación de *Agile* en el desarrollo de grandes sistemas software en el que intervienen equipos de trabajo distribuidos.

El objetivo de este trabajo es analizar los problemas que presenta la aplicación de *Agile* en las circunstancias señaladas y, a partir del estudio de diferentes *frameworks* de escalado *Agile* y teniendo en cuenta diversas aportaciones empíricas, diseñar un procedimiento adecuado que permita resolver los problemas planteados. Para concluir el trabajo, se realiza una propuesta de implantación y un conjunto de recomendaciones a tener en cuenta a la hora de poner en marcha los procedimientos diseñados en un entorno real de trabajo.

Abstract

Agile project management methodologies have meant a transformation in software development. Two of its basic pillars are fast adaptation to change and close collaboration with the customer. However, after a few years of working with Agile, it has been found that this methodology is not efficient enough under certain circumstances. Such is the case of applying Agile in the development of large software systems involving distributed teams.

This study aims to analyze the problems encountered when using Agile methodologies in the aforementioned circumstances and, on the basis of the study of different frameworks for scaling Agile and taking into account different empirical contributions, designing an appropriate procedure to resolve the problems highlighted. To conclude the report, a design for the implementation process is suggested including also some recommendations that must be considered when applying the proposed approach in a real working environment.

1. INTRODUCCIÓN Y PLANTEAMIENTO DEL PROBLEMA

1.1. Origen de las metodologías Ágiles en la gestión de proyectos

Una metodología de gestión de proyectos se define como un conjunto de métodos, técnicas, procedimientos, plantillas y recomendaciones de mejores prácticas de aplicación en la gestión y desarrollo de un proyecto (1). En los años 50 se instauraron los principios sobre los que se fundamenta la gestión moderna de proyectos. Dicha gestión se basa en la predictibilidad de los sistemas y puede describirse como orientada a procesos, basada en la documentación exhaustiva y en una planificación detallada de las tareas. Además, también asume una jerarquía de capacidades y responsabilidades dentro de las distintas tareas de un proyecto, potenciando la especialización de los trabajadores. Estos sistemas de gestión de naturaleza predictiva, han sido empleados con éxito en campos como la arquitectura, construcción civil, defensa e ingeniería.

Cuando en los años 70 comenzó a surgir la nueva industria del software, se comenzaron a utilizar este tipo de prácticas de naturaleza predictiva también para el caso de desarrollo de productos software. Los principales motivos de esta adopción fueron los siguientes:

- Las metodologías predictivas que, hasta el momento se usaban en entornos de ingeniería tradicional, habían resultado de exitosa aplicación, tal y como se ha comentado.
- La especialización y jerarquía del personal en las organizaciones encajaba bien con la formación de equipos 'tradicionales' que trabajan en base a su capacitación y rango jerárquico.
- El coste de los procesos, materiales, etc., empleados en la industria del software eran 'complicados, pesados y caros', lo cual requería de unos estrictos procesos de producción, limitando así el público objetivo.

Con el paso del tiempo y la rápida evolución del sector TIC, los procesos de fabricación se fueron abaratando y la tecnología se fue acercando a las personas. Esto incrementó la necesidad de involucrar a los clientes en el proceso de diseño y desarrollo del producto. Además, con la aparición de Internet, se hizo primordial la reducción de los tiempos de producción para poder ser competitivos.

En este entorno altamente cambiante, las metodologías tradicionales de tipo predictivo comenzaron a mostrar dificultades. Estos pesados procesos, imprimían tiempos elevados de producción, una escasa o nula interacción con el cliente y una reducida capacidad de respuesta ante los rápidos cambios que el mercado experimentaba.

Como respuesta a los problemas planteados, surgieron nuevas formas de gestión dentro de entornos software a las que se denominó '*Lightweight Methodologies*' (metodologías ligeras). Algunas de ellas fueron resultado de la experimentación y otras fueron importadas de otras industrias como la del automóvil (Toyota: método Kanban y *Lean Manufacturing*). De esta forma, surgió un ecosistema de nuevas prácticas en los procedimientos aplicados a la industria del software entre las que podemos destacar: *eXtreme Programming (XP)*, *Scrum*, *Lean Software Development*, *Feature Driven Development (FDD)*, *Pair Programming*, etc. Un hito que merece la pena señalar fue la publicación en el año 1994 del Informe del caos (*Standish group*) (2) y que se ha utilizado en múltiples ocasiones para apoyar la crisis del software. Este documento es el primero que presenta un estudio en el que se analizan las tasas y motivos de éxito o de fallo de los proyectos software. De forma resumida, en dicho informe se concluye que más del 70% de los proyectos de software fallan de alguna manera (tienen sobrecostos, se entregan tarde y sin cumplir las expectativas de los clientes o incluso son cancelados antes de concluirse). En la siguiente imagen se muestran los resultados que se incluyeron en dicho informe sobre la tasa de éxito y fracaso de proyectos software:

For purposes of the study, projects were classified into three resolution types:

- Resolution Type 1, or project success: The project is completed on-time and on-budget, with all features and functions as initially specified.
- Resolution Type 2, or project challenged: The project is completed and operational but over-budget, over the time estimate, and offers fewer features and functions than originally specified.
- Resolution Type 3, or project impaired: The project is canceled at some point during the development cycle.

Overall, the success rate was only 16.2%, while challenged projects accounted for 52.7%, and impaired (canceled) for 31.1%.

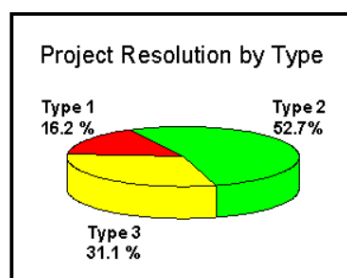


Figura 1. Informe del caos de Standish Group (1994) (2)

El término 'Metodologías *Agile*' fue acuñado en 2001 cuando un grupo de 17 expertos en desarrollo software, críticos con las metodologías tradicionales, se reunieron de manera informal en Snowbird (Utah) para tratar sobre las nuevas técnicas en el desarrollo software. Dichas técnicas fueron denominadas 'Ágiles' en contraposición a las de tipo predictivo que eran consideradas más rígidas y pesadas. En esa reunión también se establecieron los principios en los que se basan estas metodologías y que se resumieron en el 'Manifiesto Ágil'. A raíz de la introducción de dicho 'Manifiesto', las metodologías *Agile* han transformado la gestión de proyectos software promulgando valores como la tolerancia al cambio, el desarrollo iterativo e incremental, la involucración de todos los miembros de los equipos de desarrollo y la estrecha colaboración con el cliente (3).

Tras más de 20 años de uso, el éxito de estas metodologías aplicadas a la gestión de proyectos software está ampliamente demostrado (4). Sin embargo, los pilares de la metodología *Agile* se ponen en entredicho cuando se pretende usar dichas prácticas en 'proyectos de gran escala' ejecutados por diferentes equipos de trabajo que se encuentran físicamente, en muchas ocasiones, en centros de trabajo distintos (incluso en diferentes países). *Agile* se basa en el trabajo de equipo de forma autogestionada y en estrecha colaboración con el cliente y, resulta evidente, que esta colaboración es difícil de mantener en entornos de desarrollo distribuidos. Además, la implantación de metodologías *Agile* implica un importante cambio cultural en las organizaciones que deben migrar de un modelo organizacional jerárquico a una forma de trabajo donde la colaboración y la confianza sean los valores que definan las relaciones en la compañía (28).

Los grandes proyectos son difíciles de gestionar y con frecuencia no cumplen las expectativas de los clientes. Las ventajas que aporta la gestión *Agile* sobre la tradicional parecen muy atractivas, pero ¿Por qué *Agile* no resulta apropiado en entornos grandes distribuidos?

1.2. Dificultades de la aplicación de *Agile* en proyectos a gran escala

Las metodologías de trabajo asociadas a *Agile* son de fácil aplicación en entornos pequeños de trabajo. Las prácticas y ceremonias que promulga *Agile* encajan con equipos pequeños (7-9 personas) con habilidades multidisciplinarias.

En el entorno de un equipo de esas características es fácil mantener un *backlog* (listado de requisitos) único sobre el que trabajar con el *Product Owner* (cliente) para lograr evolucionar el producto de forma iterativa e incremental. Sin embargo, cuando se intenta aplicar la filosofía *Agile* a un conjunto de equipos dependientes entre sí, resulta complicado mantener la misma fluidez y flexibilidad que surge de forma natural con un único equipo.

Las dificultades más importantes que pueden encontrarse en la aplicación de *Agile* en entornos complejos están derivadas precisamente de las mismas características que le aportan 'agilidad' a la metodología de trabajo como son la independencia y autonomía de los equipos.

Durante el desarrollo de este trabajo se profundizará en las dificultades que presenta *Agile* en estos entornos. No obstante, de forma introductoria podemos destacar las siguientes:

- *Coordinación de equipos*

La autonomía y autogestión de los equipos son pilares básicos de *Agile*. Sin embargo, cuando se trabaja en grandes proyectos, se ha comprobado que dicha autogestión reduce el alineamiento entre los equipos provocando, en muchas ocasiones, ineficiencias y retrabajo (5).

- *Arquitecturas técnicas deficientes (technical debt) (15)*

En *Agile*, el orden en el que se ejecutan las tareas es decidido por el cliente. En base a dicha prioridad, el equipo es autónomo a la hora de trabajar y toma las decisiones que considera oportunas para desarrollar la tarea. Esta forma de trabajo puede originar dos problemas que impactan directamente en la calidad técnica de los desarrollos:

- Falta de priorización de tareas técnicas frente a las de negocio. El cliente será más partidario de priorizar las tareas de negocio frente a las orientadas a mejorar la arquitectura de los sistemas.
- Arquitecturas no escalables o de difícil integración con las realizadas por otros equipos trabajando en el mismo producto. Debido a la autogestión, es posible que dos equipos puedan tomar decisiones que impliquen dificultades en la integración de los sistemas o en la elección de soluciones difícilmente escalables.

Agile es una metodología que resulta muy atractiva tanto para los desarrolladores (que encuentran una forma de trabajar donde pueden potenciar su creatividad) como para los responsables de las compañías (que perciben unos procesos menos rígidos y con los cuales es posible reducir el *Time-to-Market*). Sin embargo, resulta evidente que, en ciertos entornos y, en concreto, en grandes proyectos, es necesario añadir elementos que permitan coordinar y alinear a los diferentes equipos entre sí y con la estrategia corporativa.

1.3. Metodologías híbridas de gestión de proyectos

Teniendo en cuenta los problemas anteriores, tanto desde el campo empresarial como desde entornos académicos, están proliferando experiencias y/o nuevos marcos de trabajo que ofrecen soluciones a estas nuevas necesidades emergentes. En realidad, la solución no es nueva. Muchas de las metodologías de gestión de proyectos, bien sean de carácter predictivo como de tipo *Agile*, definen recomendaciones y buenas prácticas, pero insisten en que cada caso es diferente y que, en muchas ocasiones es preciso realizar adaptaciones (*tailoring*) de las prácticas teniendo en cuenta las condiciones particulares de cada proyecto o cada entorno empresarial. Estas adaptaciones están originando, en ciertos casos, la aparición de metodologías mixtas (o híbridas) de gestión de proyectos en las que conviven elementos propios de prácticas diversas.

De esta forma, asociado al término ‘Escalado *Agile*’ pueden encontrarse un conjunto de nuevos marcos de trabajo (*frameworks*) que, sobre una base *Agile*, incorporan mecanismos de colaboración y alineamiento ‘multiequipo’ con el objetivo de minimizar los inconvenientes que *Agile* presenta en entornos grandes (6).

En este trabajo se realizará una revisión de dichas metodologías, analizando de forma más amplia las de uso más extendido en la actualidad.

2. OBJETIVOS Y PLAN DE TRABAJO

Considerando la situación planteada, este trabajo profundiza en los principales problemas que presenta el paradigma *Agile* cuando se pone en práctica en el ámbito de la gestión de grandes proyectos software con equipos de trabajo distribuidos. El objetivo es proponer opciones para mitigar dichos problemas, lo cual permite acometer esta tipología de proyectos incrementando su probabilidad de éxito. El trabajo se organiza en las siguientes secciones:

- Desarrollo del estado del arte sobre metodologías de gestión de proyectos. Se realiza un completo estudio que incluye los siguientes puntos:
 - Presentación del origen de la gestión de proyectos y sus distintas tendencias (predictiva vs *Agile*). Esta sección considera también las ventajas e inconvenientes de ambas aproximaciones así como el ámbito de aplicación de cada una de ellas.
 - Estudio de la situación actual de las metodologías *Agile* aplicadas para el caso de grandes proyectos software. Se presentan los marcos de trabajo más populares incluyendo un estudio comparativo y el grado de implantación de cada uno de ellos.
 - Presentación de las herramientas y técnicas de uso más frecuente en entornos *Agile* y, en concreto, en grandes proyectos.
- Diseño de una metodología híbrida de gestión de proyectos orientada a proyectos de desarrollo software con equipos de trabajo distribuidos. Para seleccionar los elementos que formarán parte de la metodología, se tendrán en cuenta diferentes aspectos como la tipología de proyectos a gestionar, las características de las metodologías presentadas y las ventajas e inconvenientes que presentan dichas metodologías cuando se aplican a ese tipo de proyectos.
- Elaboración de recomendaciones para la implantación del proceso.
- Conclusión del trabajo incluyendo una revisión de los resultados y concretando las posibles acciones futuras que podrían derivarse del trabajo presentado en este estudio.

3. ESTADO DEL ARTE

3.1. Gestión de proyectos: antecedentes

3.1.1. Gestión de proyectos tradicional

De manera sencilla puede considerarse que el propósito principal de la gestión de proyectos es la de coordinar todas las actividades y los recursos asignados al mismo para conseguir el objetivo del proyecto.

La idea básica que subyace bajo el enfoque tradicional de gestión de proyectos es éstos (ya sean simples o complejos) se comportan de una forma predecible. Además, resulta sencillo establecer límites claros en el desarrollo, lo cual permite definir un plan concreto y seguir ese plan sin esperar muchos cambios.

Mediante la gestión de proyectos tradicional se persigue la optimización del trabajo gracias a la definición y seguimiento de un plan definido que permita cumplir con el alcance en el tiempo establecido y con los costes presupuestados. En este tipo de enfoque, denominado también predictivo, el proyecto se gestiona en base a las siguientes fases básicas (1):

1. Iniciación
2. Planificación
3. Ejecución
4. Monitorización y control (proceso continuo a lo largo del proyecto)
5. Cierre

Dichos pasos son ejecutados de forma secuencial (excepto el de monitorización y control, que es un proceso continuo) y uno de los valores fundamentales en los que basa su éxito es la ‘anticipación’ y la predictibilidad. Se promueve el seguimiento y control de un plan, la documentación exhaustiva y la definición de un alcance cerrado que, en conjunción con el tiempo y el coste de desarrollo, definirán el éxito o no del proyecto.

En la siguiente imagen se muestra un ejemplo de ciclo de vida predictivo donde algunas de las fases enunciadas anteriormente se dividen en varias.

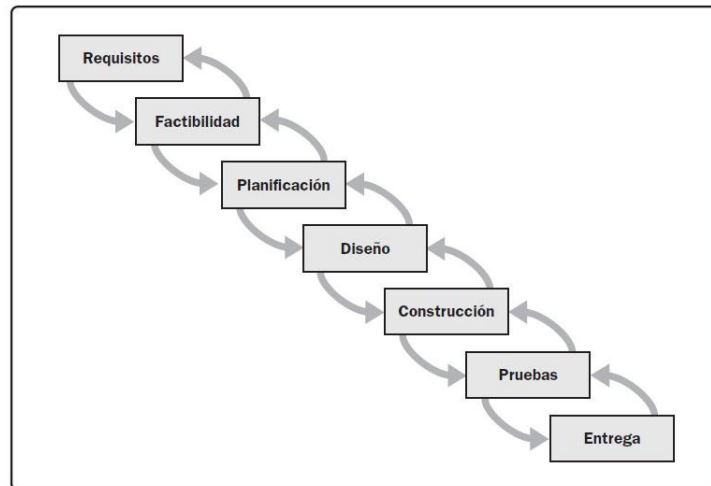


Figura 2. Ejemplo de ciclo de vida predictivo
 Fuente: Guía del PMBoK Quinta edición (Project Management Institute)

En los proyectos gestionados de esta forma se establece una dependencia entre el alcance, tiempo y coste de un proyecto que puede representarse mediante un triángulo (triple restricción). Esta es la forma en la que se representa que, cualquier variación en uno de los elementos de la tripla impacta en los otros dos. En concreto, en este enfoque, el alcance se cierra al comienzo del proyecto de manera formal (incluso a veces a través de un contrato entre las partes) provocando en la mayoría de las ocasiones, como se ha comentado, variaciones en las otras dos variables (coste y tiempo de entrega).

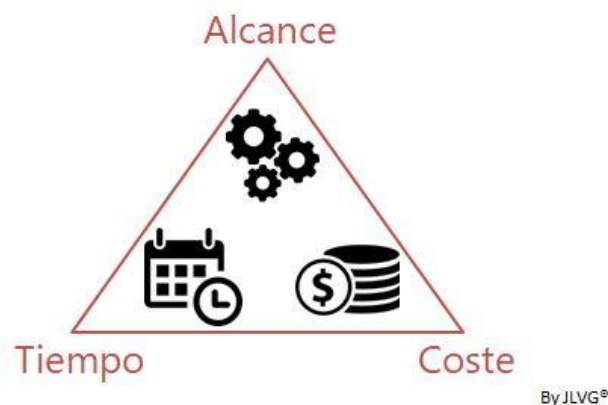


Figura 3. Triple restricción para entornos predictivos
 Fuente: <https://managementplaza.es>

Dentro de este enfoque, el modelo más implantado es el del *Project Management Institute (PMI)* que define en el PMBoK (*Project Management*

Body of Knowledge) (1) los fundamentos, estándares, pautas y normas reconocidas como ‘mejores prácticas’ para la gestión de proyectos.

A pesar de que, en este enfoque, se resalta como ventaja principal que es aplicable en cualquier tamaño y tipología de proyectos, se está comprobando en estas últimas décadas que, en muchos casos, su aplicación presenta deficiencias debido a su excesiva rigidez, lentitud y el alto coste asociado al cambio.

3.1.2. Gestión de proyectos Agile

Las limitaciones de los enfoques tradicionales de gestión de proyectos impulsaron la introducción de nuevos enfoques en este campo. Dichos enfoques, englobados bajo la denominación *Agile*, se basan en favorecer la ‘adaptabilidad’ a los cambios durante el ciclo de vida del proyecto en contraposición a la ‘anticipación y predictibilidad’ promulgada por las metodologías tradicionales. Los cambios en el entorno actual son inevitables y frecuentes, por lo que se promueve la ‘flexibilidad’ y se reconoce que no es posible definir un plan estable que sirva para dirigir el proyecto desde sus comienzos.

Estas ideas comenzaron a ganar popularidad y a materializarse en metodologías concretas a partir de la publicación del ‘Manifiesto *Agile* (2001) (3)’. Este Manifiesto fue escrito por un grupo de autores que, preocupados por la situación del desarrollo software, definieron los siguientes valores como directrices básicas de las metodologías *Agile*:

Manifiesto por el Desarrollo Ágil de Software

*Estamos descubriendo formas mejores de desarrollar software tanto por
Estamos descubriendo formas mejores de desarrollar software tanto por
nuestra propia experiencia como ayudando a terceros. A través de este
trabajo hemos aprendido a valorar:*

Individuos e interacciones sobre procesos y herramientas

Software funcionando sobre documentación extensiva

Colaboración con el cliente sobre negociación contractual

Respuesta ante el cambio sobre seguir un plan

Esto es, aunque valoramos los elementos de la derecha, valoramos más

La filosofía *Agile* promueve la entrega parcial y regular del producto final a través de ciclos de desarrollo iterativos e incrementales. El cliente puede introducir cambios en el proyecto incluso en etapas avanzadas. Dichos cambios son priorizados en función del **valor** que aportan al proyecto. No existe una estructura jerárquica dentro de los equipos, sino que el desarrollo es ejecutado por un equipo multidisciplinar en el que todos los integrantes colaboran para obtener el mejor resultado posible de un proyecto (7,8).

En *Agile* la 'triple' restricción cambia de orientación. En este caso el alcance no se determina de forma cerrada sino que se va adaptando a medida que el proyecto avanza. En este caso, el tiempo y el coste del proyecto son variables que se priorizan frente al alcance.

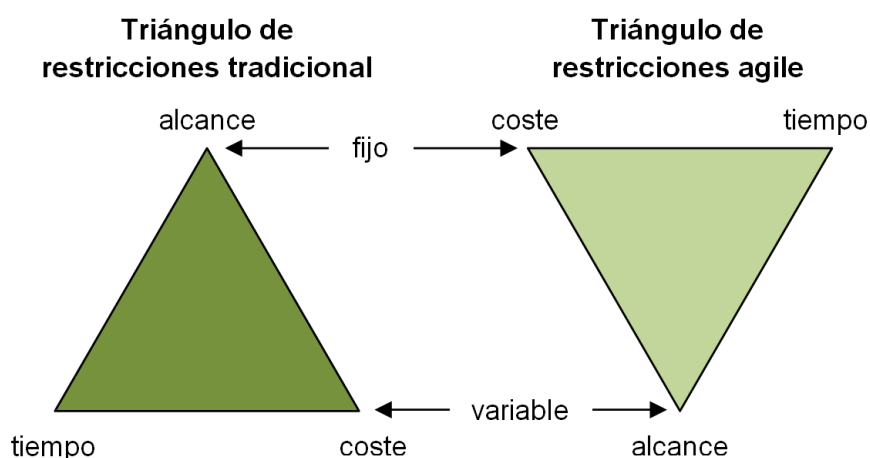


Figura 4. Triple restricción de enfoque predictivo vs Agile (7)

En un proyecto gestionado mediante *Agile*, se definen tres roles básicos que intervienen en el proceso (9,10):

- *Product Owner*. Actúa como representación del cliente y él es quien decide cuales son las características funcionales del producto.
- *Equipo Agile (Team)*. Es el equipo de desarrolladores multidisciplinar, integrado por programadores, diseñadores, arquitectos, etc., que, de forma auto-organizada, se encargará de desarrollar el producto.
- *Facilitador*. Este rol se encarga de fomentar e instruir sobre los principios ágiles dentro del proyecto y de la organización.

El proyecto evoluciona en ciclos incrementales de desarrollo siguiendo un proceso 'Iterativo e Incremental'. Dichos ciclos de desarrollo son cortos (entre 2 y 4 semanas de duración) y, al principio de cada ciclo se define el alcance y objetivo del mismo, que no puede ser modificado en el tiempo que dura el ciclo. Al final de dicho ciclo se debe disponer de un conjunto de nuevas funcionalidades para el producto que le provean valor añadido de forma inmediata.

Además, *Agile* se inspira en la filosofía *Lean*, que se centra en la búsqueda continua del valor para el cliente, eliminando todo lo que no contribuye a ese valor (waste) y que impacta negativamente en el tiempo y coste de producción. Como herramienta para optimizar el flujo de trabajo para la generación de valor, surge el tablero *Kanban* como herramienta que aporta visibilidad del trabajo diario ayudando a eliminar los impedimentos que puedan ralentizar ese flujo continuo (cuellos de botella, demasiado trabajo en paralelo, etc.).

Dentro del enfoque *Agile* se han promovido diversas metodologías concretas entre las que podemos destacar *Scrum* (9,30) (orientado a la gestión del proyecto) y *eXtreme Programming* o *XP* (10) (orientados al desarrollo software). De entre ellas, *Scrum* es la de más amplio uso. En la siguiente imagen se muestra el proceso *Scrum*:

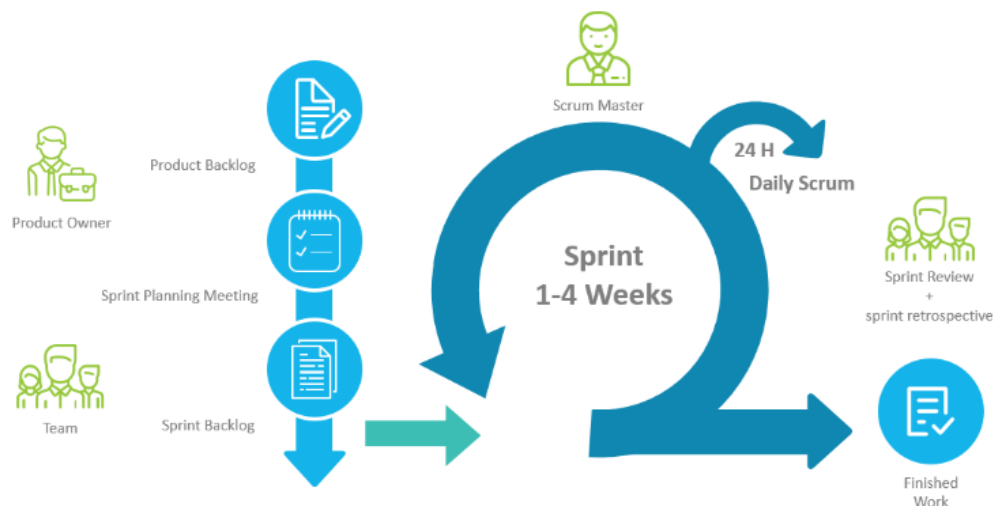


Figura 5. Representación del proceso Scrum
Fuente: <https://www.powerslides.com>

El proceso *Scrum* se desarrolla en las siguientes fases:

- Se dispone de un *backlog* de producto o listado de funcionalidades (historias de usuario) que pueden considerarse como los requisitos que debe satisfacer el producto. Dichas funcionalidades se ordenan según su prioridad que debe estar alineada con ‘el valor’ que cada una aporta al producto. La ordenación de ese *backlog* corre a cargo del *Product Owner* (representación del cliente).
- Durante el *sprint planning meeting* (reunión de planificación del *sprint*), el *Product Owner* describe las funcionalidades de más alta prioridad al equipo. El equipo obtiene de esa reunión toda la información que necesita para transformar y dividir dichas historias de usuario en tareas más detalladas que serán implementadas por el equipo. El objetivo de dichas reuniones no es el de analizar de forma completa el alcance de todo el contenido del *backlog*, sino que se centra en las tareas de más alta prioridad. Como resultado de dicha reunión, se obtiene el listado de historias que serán acometidas durante el siguiente ciclo de desarrollo (*sprint*).
- Durante la ejecución del *sprint* el equipo trabaja de forma autónoma y auto-organizada. Son los propios miembros del equipo los que seleccionan las tareas a realizar y se coordinan mediante reuniones breves diarias (*daily*s).

- El último día del *sprint* se realiza el *Sprint review*, que consiste en la presentación de los resultados del trabajo al *Product Owner*. El objetivo es asegurar que el trabajo realizado se corresponda con las expectativas del cliente. En caso que no sea así, se plantearán cambios que podrán ser acometidos en los siguientes ciclos de trabajo. Idealmente, el trabajo realizado durante el *sprint*, podría ser desplegado en producción de manera inmediata dotando así de cambios al producto de forma continua.
- Para cerrar el *sprint* se celebra una reunión denominada 'Retrospectiva' en la que el equipo analiza su trabajo con el fin de aprender de lo que se ha hecho bien y corregir aquello que es preciso mejorar en futuras iteraciones.

Agile actualmente se considera mucho más que un *framework* de gestión de proyectos. Se considera prácticamente una filosofía de trabajo y la adopción real de *Agile* necesita de un cambio profundo de mentalidad en las organizaciones. Es necesario pasar de un modo 'command and control' (organizaciones funcionales verticales) a un modo 'colaborativo, basado en la confianza y con orientación al cliente' (organizaciones horizontales orientada a procesos y productos).

3.1.3. *Ventajas, inconvenientes y aplicaciones de cada enfoque*

Ambas aproximaciones son válidas a la hora de gestionar proyectos y, a priori, no es posible afirmar que una metodología es mejor que la otra (11). La selección de una metodología concreta a emplear debe realizarse teniendo en cuenta la naturaleza del proyecto que va a gestionarse.

En general, en la literatura se afirma que la metodología tradicional es más apropiada para proyectos con objetivos claros y con muy poco nivel de incertidumbre (11,12). En dichos proyectos, se espera que no exista un gran número de cambios y que no sea preciso una alta involucración del cliente durante el desarrollo del proyecto. En esos entornos, se enfatiza la existencia del seguimiento de un plan que sirva para medir el avance del proyecto. Además, también se recomienda utilizar estas metodologías cuando se aborda la gestión de grandes proyectos (bien en cuanto a la complejidad o cuanto al tamaño) o en el caso de proyectos críticos.

Por otra parte, la gestión de proyectos *Agile*, se orienta más a aquellos que tienen una filosofía creativa y/o innovadora donde hay un alto grado de incertidumbre o bien en aquellos casos en los que los requisitos están incompletos y se espera un alto volumen de cambios (11,12). Debido a estas características, la gestión del proyecto se organiza en torno a un modelo en el que el desarrollo se plantea de forma iterativa e incremental con frecuentes modificaciones y actualizaciones. Este modelo requiere también de la colaboración continua entre los miembros del equipo y con el cliente. Es por tanto que, en principio, estas metodologías están recomendadas para el caso de gestión de proyectos software pequeños o medianos cuyos equipos se encuentren ubicados en una misma localización física.

Agile es igual a colaboración y contacto directo con el cliente. Para el caso de grandes proyectos software esta comunicación se complica y comienzan a aparecer dependencias entre distintos equipos de trabajo que deben ser resueltas. Varios investigadores (13,14) consideran la '**comunicación**' y '**coordinación**' entre equipos como uno de los retos más importantes a superar en estos entornos. Según (13), en las grandes organizaciones, la comunicación informal y la gestión simple de *backlogs* por equipo, no es suficiente para realizar una efectiva gestión de los requisitos. En las grandes organizaciones, la estrategia debe fluir desde los niveles más altos de la organización hasta los equipos individuales; y dentro de este nivel, la estrategia debe ser compartida entre todos los equipos con el objetivo de alinear el trabajo con la misma.

Una vez definido el trabajo a realizar e incluso habiendo conseguido alinear los diferentes *backlogs* de los equipos, surge el problema de la '**gestión de las dependencias**'. En las metodologías tradicionales en cascada, la existencia de una planificación detallada controlada por el Jefe de Proyecto permite gestionar esas dependencias. En *Agile* en cambio se promulga la 'autogestión de los equipos'. En principio cada equipo es autónomo a la hora de acometer las funcionalidades seleccionadas para el *sprint* e, incluso habiendo alineado los *backlogs* de los distintos equipos, esta autonomía puede ocasionar cuellos de botella por la aparición de dependencias no planificadas o derivadas del análisis más laxo de los requisitos que se implementa con *Agile*. (13–15).

Otro tema en el que muchos autores coinciden es la posible '**deficiencia técnica de arquitectura (technical debt)**' que puede derivarse de la filosofía *Agile*.

- Según (15), dejando en exclusiva al cliente (*Product Owner*) la capacidad de decidir qué requisitos se acometen, se corre el riesgo de que solamente se prioricen los desarrollos orientados al negocio en detrimento de los relacionados con la optimización y mejora del software.
- Además, el mismo autor también expone que la creación de equipos generalistas (multidisciplinares) cuando se trabaja en el desarrollo de elementos software complejos, no es eficiente. En muchos casos, ciertos sistemas requieren de años de experiencia para ser completamente comprendidos y, en esos casos, resulta más recomendable la creación de ciertos equipos especializados frente a equipos multidisciplinares menos experimentados.

Por último, en la literatura analizada se plantea el problema de la '*agilidad* de las grandes organizaciones'. En una pequeña empresa, donde un proyecto se gestiona con un equipo de forma Ágil estando además en comunicación directa con el responsable de la compañía, es posible implantar *Agile* directamente sin ningún tipo de problemas o restricciones. Sin embargo, cuando hablamos de grandes empresas, aparecen barreras importantes a la hora de aplicar *Agile*. Según (16), en estos casos se presentan limitadores organizacionales importantes como pueden ser el diseño de la organización (demasiado jerárquico) y la dificultad a la hora de establecer mecanismos ágiles para la toma de decisiones.

3.2. Escalado *Agile*

3.2.1. Metodologías Híbridas de gestión de proyectos

Una metodología híbrida de gestión de proyectos es aquella que implementa prácticas pertenecientes a distintos modelos de trabajo. Como ya se ha comentado, esta idea no es nueva, ya que cualquier metodología promulga la necesidad de 'tailorización' (adecuación de los procesos a las necesidades propias) y esto puede verse como una forma de hibridación.

La implantación de *Agile* en entornos software está suponiendo una revolución. Son muchas las empresas que están viendo en *Agile* una forma de

acelerar sus procesos de desarrollo y entrega del producto, lo cual impacta directamente en una mayor satisfacción por parte del cliente. Sin embargo, en algunas ocasiones no es posible la aplicación de metodologías *Agile* en estado puro, bien sea por las características de los proyectos a acometer o porque las compañías no se encuentran todavía en un estado de madurez tal como para abordar importantes cambios estructurales. La combinación de técnicas de diferentes metodologías de gestión de proyectos es cada vez más frecuente (29). Incluso PMI-PMBok, metodología tradicionalmente de naturaleza predictiva, ha incorporado en su última versión (Guía PMBoK, sexta edición (1)) referencias a prácticas híbridas y *Agile*. Además, esta versión también incorpora una guía escrita en colaboración con la *Agile Alliance* (17) donde se proponen técnicas y herramientas concretas orientadas a entornos híbridos *PMBok-Agile*.

Derivados de esta tendencia, en los últimos años han surgido diferentes *frameworks* que proponen formas de trabajo híbridas. Este trabajo se centra en las orientadas a gestión de proyectos a gran escala.

Según diversas fuentes (18,25) los *frameworks* de escalado *Agile* de uso más extendido son:

- Scrum of Scrums (Jeff Sutherland) (20)
- LeSS - Large Scale Scrum (Larman/Vodde) (21)
- DAD - Disciplined Agile Delivery (Ambler/Lines) (22)
- Método usado en Spotify (Kniberg) (24)
- SAFe - Scaled Agile Framework (Leffingwell) (23,26)

Los anteriores marcos de trabajo se construyen sobre la base de un trabajo en modo *Agile* a nivel de equipo a partir de la cual aportan soluciones que permiten coordinar el trabajo de múltiples equipos trabajando en paralelo. El objetivo es asegurar la adecuada integración de todas las aportaciones de forma que se puedan cumplir los objetivos de un gran proyecto evitando retrasos y deficiencias en los desarrollos.

A continuación, se describen brevemente los *frameworks* de escalado de más amplio uso en la actualidad.

Scrum of Scrums (Jeff Sutherland) (20)

Creado por uno de los padres de *Scrum*, este *framework* facilita el alineamiento de los equipos *Scrum* mediante la incorporación de una reunión adicional de coordinación entre representantes de los equipos. Dichas reuniones son similares a las *daily* de equipo, pero en las que se plantea además de las cuestiones habituales, la identificación de posibles dependencias.

De la misma forma que para los equipos *Scrum* existe un límite máximo de participantes en estas reuniones, en *Scrum of Scrums* también se establece un número máximo de equipos de segundo nivel para los que se aplica el mismo límite (entre 5 y 9 equipos). En caso de que se supere este límite, se añadiría una nueva capa para el escalado. En la figura siguiente se representa gráficamente la organización de un equipo de proyecto con 3 niveles *Scrum*.

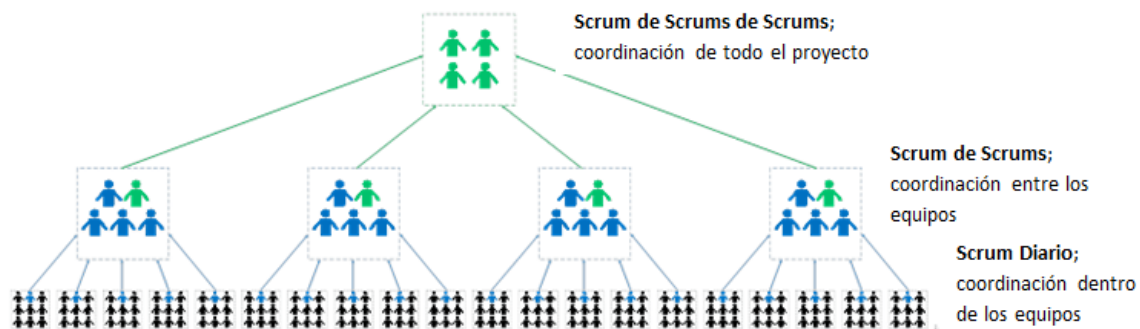


Figura 6. Distribución de equipos para Scrum de Scrums
Fuente: <http://managementplaza.es>

Esta forma de escalado es la más sencilla de las planteadas y presenta como inconveniente principal sus limitaciones cuando se intenta aplicar a estructuras realmente complejas.

LeSS - Large Scale Scrum (Larman/Vodde) (21)

Según sus creadores, LeSS es un *framework* que se encuentra basado principalmente en la experiencia. La base teórica es simple y consiste en una forma de aplicar los principios, reglas, elementos y propósito de *Scrum* en un contexto a gran escala de la forma más sencilla posible.

En LeSS no se añaden roles ni artefactos adicionales a los propios de *Scrum*, lo que evita incrementar la carga que ya tienen los propios equipos o dificultar la comunicación directa entre el cliente y el equipo (como puede ocurrir, como se verá, en otros *frameworks* como SAFe)

Es un modelo basado en filosofía Lean (7,19) y que debe partir desde la organización en sí misma. No se consigue absolutamente nada implementando *Scrum* a nivel de equipo si la organización continúa siendo dirigida de forma jerárquica basada en estructuras funcionales.

LeSS ofrece dos *frameworks* dependiendo del tamaño de la organización que debe acomodar

- LeSS. Para 2–8 equipos.
- LeSS Huge. Más de 8 equipos.

Ambas metodologías se basan en los siguientes elementos:

- Se mantiene un único *Product Owner* y un *backlog* de producto.
- La ejecución del *sprint* se encuentra sincronizada en todos los equipos.
- Se genera un incremento común para el producto derivado del trabajo de todos los equipos.

En la siguiente imagen muestra un resumen de los elementos del *framework* y sus relaciones.

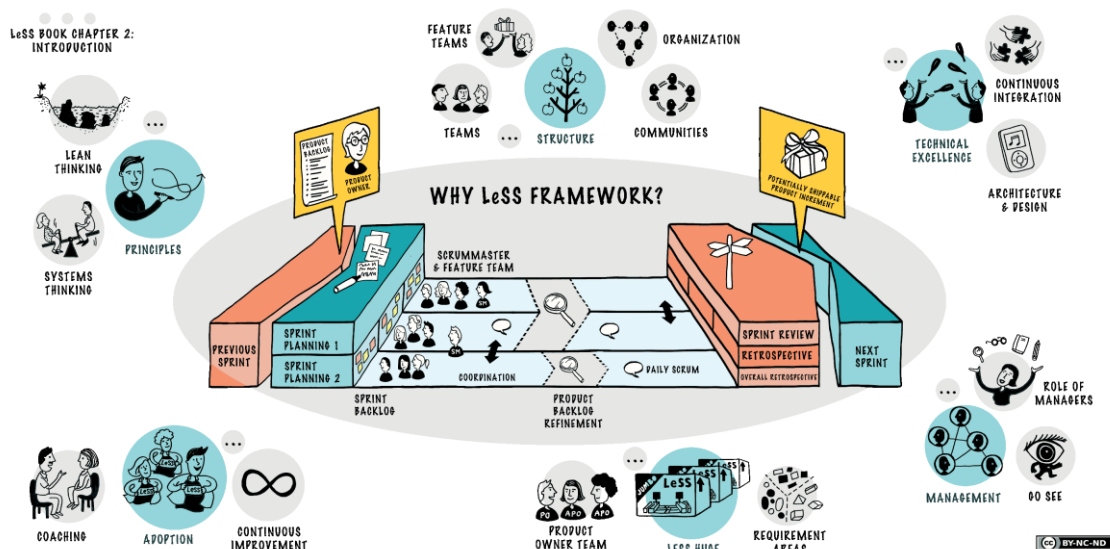


Figura 7. Representación gráfica del framework LeSS (21)

Este *framework* es el más fiel a *Scrum*, ya que intenta reutilizar las herramientas de las que dispone esa metodología con un alcance mayor que el

del simple equipo. Su principal inconveniente viene derivado precisamente de esa característica, ya que el implementar dichas prácticas a nivel más amplio puede presentar dificultades importantes en las organizaciones con estructuras verticales más tradicionales.

DAD - Disciplined Agile Delivery (Ambler/Lines) (22)

DAD, aunque no se define propiamente como un modelo orientado al escalado de prácticas *Agile*, es una metodología escalable. Sus autores insisten en definir a DAD como un '*toolkit*' en vez de un '*framework*'. Con esto se pone de relieve la naturaleza poco prescriptiva de DAD que ofrece diferentes opciones a la hora de abordar la gestión de proyectos en vez de prácticas concretas. Según sus propios autores, DAD es un conjunto de herramientas (toolkit) basado en principios ágiles pero que incluye aportaciones de un amplio conjunto de metodologías.

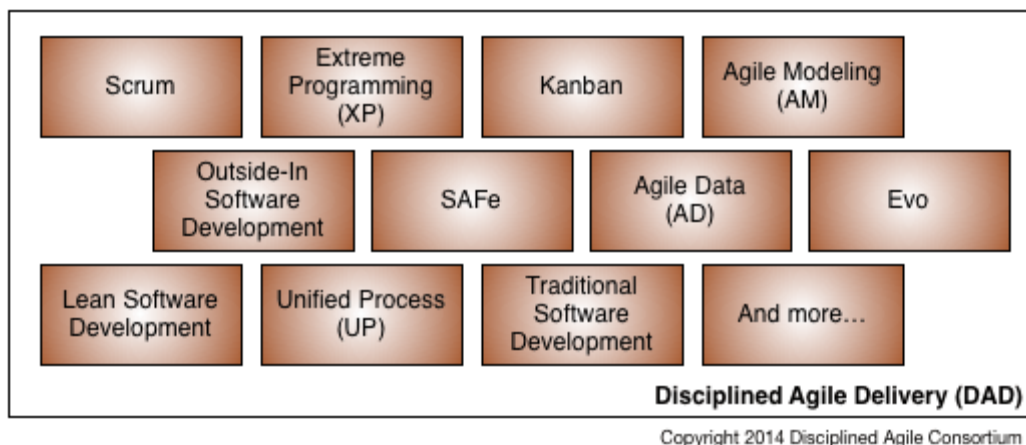


Figura 8. Herramientas que aportan diversos elementos a DAD (22)

DAD se enmarca dentro de un toolkit más completo (DA – Disciplined Agile) orientado al desarrollo completo del producto que incluye cuatro capas:

- Disciplined Agile Delivery (DAD)
- Disciplined DevOps
- Disciplined Agile IT (DAIT)
- Disciplined Agile Enterprise (DAE)

DAD propone un método 'dirigido por el valor y el riesgo: risk-value driven' en el que no solo es necesario tener en cuenta el valor aportado a la organización, sino que también es preciso considerar los posibles riesgos de,

por ejemplo, crear una arquitectura deficiente. Teniendo esto en cuenta, DAD propone establecer un ciclo de vida completo para el producto (desde la concepción de la idea hasta su despliegue).

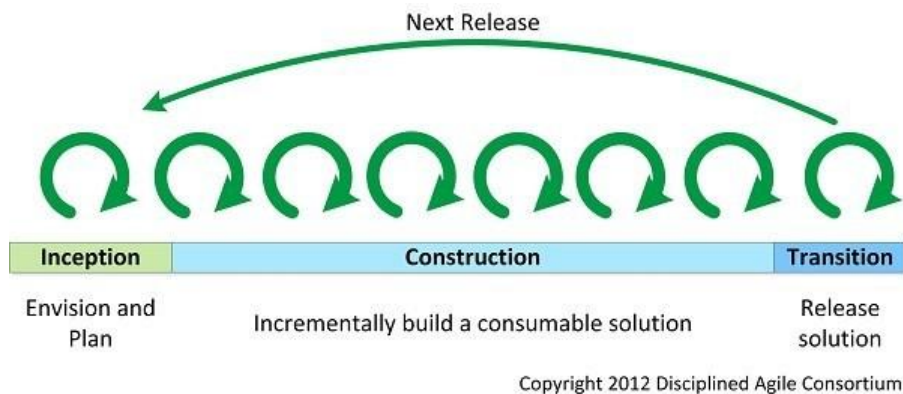


Figura 9. Vista de alto nivel del ciclo de entrega DAD (22)

El principal inconveniente que presenta DAD es la falta de concreción en los métodos a utilizar, lo cual viene derivado precisamente de su naturaleza poco prescriptiva.

Método usado en Spotify (Kniberg) (24)

Spotify es una empresa sueca dedicada a la reproducción de música vía streaming. Desde el comienzo de su andadura adoptó el uso de metodologías ágiles como forma de trabajo. Sin embargo, pronto se dieron cuenta de que el modelo no facilitaba el alineamiento de todos los equipos frente a una estrategia común, así que desarrollaron un conjunto de conceptos y estructuras no jerárquicas que les permitió organizar los grupos de trabajo.

Spotify se basa en una metodología *Scrum* a nivel de equipo y, como tal, parte con equipos multidisciplinares y auto-organizados de 7/9 personas que denomina *Squad* (escuadrón). A cada *Squad* se le asigna un ámbito de responsabilidad que es conocido por toda la organización.

Los *Squad* se agrupan en estructuras mayores según ámbitos relacionados. Estas agrupaciones se denominan *Tribes* (tribus) y trabajan en áreas relacionadas entre sí. El tamaño de cada tribu puede oscilar desde las 40 hasta las 120 personas. El tamaño máximo no ha sido seleccionado con criterios arbitrarios, sino que está basado en el concepto del *número de Dumbbar* (26) que afirma que las personas no pueden mantener una relación social simultánea con más de 150 personas. Por encima de ese límite se comienzan

a establecer reglas, burocracia y capas adicionales que facilitan la comunicación, pero suponen una pérdida de tiempo. Además, las tribus celebran reuniones periódicas donde se muestran los avances, se comparten experiencias y se fomenta la cultura de compartir el aprendizaje.

De forma adicional, las personas con intereses similares se agrupan en *Chapters* (capítulos) y *Guilds* (gremios). Ambas estructuras están orientadas a incluir los beneficios de economía de escala dentro de las organizaciones *Scrum* posibilitando que las personas con tareas similares o con intereses comunes intercambien información de forma que se cree un conocimiento común y una cultura de empresa.

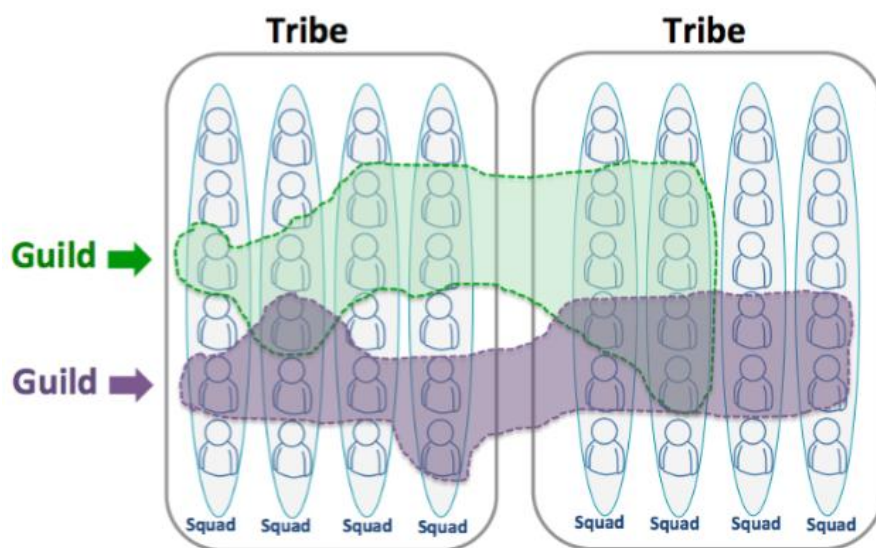


Figura 10. Método de escalado Agile en Spotify (24)

El caso planteado por *Spotify* puede considerarse más como un caso de éxito asociado a una 'tailorización' o adaptación particular de *Scrum* que como un *framework* propiamente dicho.

SAFe - Scaled Agile Framework (Dean Leffingwell) (23,26)

SAFe se encuentra también orientado a establecer procedimientos que permitan a las empresas implantar la agilidad en todos sus niveles. Es un *framework* que se sustenta sobre principios *Lean-Agile* y considera tres niveles de abstracción con *backlogs* propios que es preciso coordinar: equipo, programa y portfolio.

- *Nivel de equipo.* El trabajo a este nivel se organiza en modo *Scrum* con un *backlog* propio. La coordinación con los niveles superiores se realiza a través del *Scrum Master* o el *Product Owner*.
- *Nivel de programa.* Aparecen nuevos roles como el *Release Train Engineer (RTE)*, *Product Manager* o *System Architect*. Estos roles se encargan de asegurar que los equipos estén alineados en todos los ámbitos: tanto el estratégico como el técnico. Se trabaja en base a un *backlog de programa* que contiene una lista priorizada de características a implementar (*feature o de negocio; enabler o técnica*). Para coordinar a todos los equipos que están involucrados en el desarrollo se definen los *Agile Release Train (ART)*. Los ART permiten establecer mecanismos de sincronización entre equipos con el objetivo de generar resultados de forma coordinada con una cadencia constante y predefinida. La evolución del producto obtenida en ese periodo de tiempo se denomina *PI (Product Increment)* y suele tener una duración de entre 8 y 12 semanas (4-6 *sprints*). Es habitual (aunque no obligatorio) que, tras cada PI, se genere una nueva *Release* (funcionalidad desplegada al cliente).
- *Nivel de portfolio.* En este nivel se trabaja en base a *Épicas* (historias de usuario de gran alcance) que definen los objetivos de negocio y las directrices a nivel de arquitectura de los sistemas. Estas *Épicas* fluyen a los niveles inferiores consiguiendo alinear los niveles de programa y equipo con el portfolio (estrategia).

La siguiente gráfica muestra una representación del *framework* completo.

SAFe, a pesar de ser el de uso más extendido en la actualidad, es un *framework* no exento de críticas. Dichas objeciones vienen sobre todo argumentadas por los sectores más críticos con las metodologías en cascada y más afines al agilismo. A SAFe se le tacha de ser no ser una metodología lo suficientemente flexible y que, por lo tanto, no es muy fiel a la filosofía *Agile*. Además, también contempla la existencia de roles especializados que no forman parte de los equipos, lo cual implica una pérdida de parte de la autonomía de la que los equipos disponen. No obstante, a pesar de estos inconvenientes, puede afirmarse que es un marco de trabajo que plantea soluciones para la gestión de grandes proyectos con metodologías Ágiles.

Comparativa de Frameworks

A modo de breve resumen, se incluye en la siguiente tabla una comparativa de algunos de las características principales de los *frameworks* de escalado que se han descrito.

	Scrum de scrums	Less	DAD	Spotify	SAFe
Framework (equipo)	Scrum	Scrum	Scrum/Kanban	Metodología propia al estilo Scrum	Scrum/Kanban/Lean
Coordinación / Control	Control distribuido con métodos ligeros de coordinación	Priorización centralizada. Coordinación distribuida	Directrices centralizadas. Coordinación distribuida	Control y coordinación distribuido	Priorización y directrices centralizadas (formal). Coordinación distribuida
Elementos diferenciadores del framework	Método simple de implantar	Alineamiento basado en un único Product Owner. Método no prescriptivo.	Método centrado en asegurar la calidad técnica de las soluciones	Método muy ágil y con poca sobrecarga de funciones y ceremonias	Método muy completo y en constante evolución que contempla toda la actividad empresarial.
Riesgos o debilidades	Escasa documentación y recomendaciones de implantación. No parece apropiado para muy grandes entornos	Extiende las prácticas scrum de forma 'pura' lo que genera dificultad en organizaciones muy jerarquizadas y tradicionales	Recomendaciones no concretas en cuanto al 'cómo' lo cual puede dificultar la implantación	No es prescriptivo. Define roles y funciones transversales de coordinación pero no especifica el 'cómo' articular las relaciones	Framework complejo que requiere de formación para implementarlo adecuadamente. Se percibe como demasiado prescriptivo.
Coste de implantación	Bajo	Bajo/Medio	Medio/Alto	Bajo	Alto

Tabla 1. Resumen comparativo de métodos de escalado Agile

3.2.2. Experiencias de hibridación

En los últimos años un gran número de empresas se han embarcado en la adopción de metodologías híbridas de gestión de proyectos. Según el ‘13th State of Agile Survey’ publicado en mayo de 2019 (25) (mayor y más antigua encuesta acerca del uso de métodos *Agile* en el campo del desarrollo software), SFe es el *framework de escalado* de uso más extendido. Dicha encuesta se lleva a cabo con periodicidad anual y recoge los siguientes resultados en esta última edición:

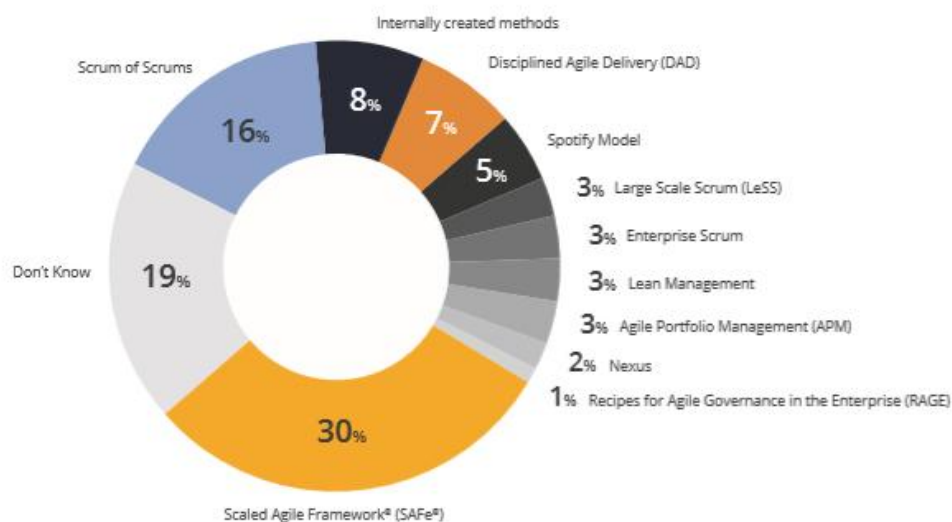


Figura 12. Métodos de escalado más usados (Mayo 2019) (25)

Dicha información coincide con la que publican diferentes autores (18,27) en la cual queda patente el predominio de SFe en este campo. En (18) los autores publican un informe muy completo sobre las ventajas y dificultades en la implantación de SFe (publicado en Nov-18). En dicho estudio se afirma que, aunque sí existe material en el que se profundiza en las causas de éxito en la implantación de estas metodologías, actualmente no se han publicado estudios comparativos, lo cual es sorprendente, ya que es un tema que está generando bastante interés. En el estudio del arte realizado en este trabajo se corrobora la opinión de los autores, ya que no se ha podido obtener gran información en este sentido. En el caso del informe publicado en (18) los autores, dada la falta de artículos científicos sobre este tema, toman como

fuentes de información tanto literatura académica como literatura gris¹. Aunque en teoría, las revisiones sistemáticas de literatura para el caso de Ingeniería del software (systematic literature review, SRL) deberían incluir todo tipo de fuentes de información (incluyendo la literatura gris) esta práctica no es frecuente en ámbitos académicos con lo que se pierde la opinión de los profesionales que no publican en dichos foros. Teniendo esto en cuenta, para el análisis de la situación de los distintos *frameworks* de hibridación, se tomarán distintas fuentes incluyendo también literatura no convencional. El objetivo es realizar un pequeño estudio, similar al realizado en (27) para SAFe, con el resto de *frameworks* presentados. A continuación se resumen los resultados obtenidos.

Experiencias con SAFe

En (18) los autores elaboran un estudio sobre los beneficios obtenidos y los retos planteados a la hora de implantar SAFe, para el cual toman como fuente de información tanto bases de datos científicas como literatura gris. Las bases de datos científicas consultadas, son las siguientes:

- Scopus
- Web of Science
- ACM
- IEEEExplore

Tras el análisis de las coincidencias obtenidas a partir de las búsquedas en las fuentes anteriores (descartando referencias duplicadas y seleccionando solo aquellas que realmente tratan sobre el framework SAFe), se obtiene el siguiente resultado:

- Hay un total de *6 artículos* en bases de datos científicas
- En el sitio web de SAFe se reportan *46 casos de estudio*
- Se encuentran *36 publicaciones* de otro tipo (presentaciones, enlaces o informes descargables en la web)

Por lo tanto, el número de artículos disponibles para el análisis es un total de 88, de los cuales, solo el 6,8%, provienen de fuentes científicas. Además, una

¹ Grey literature is defined as, “(the literature), produced on all levels of government, academics, business and industry in print and electronic formats, but which is not controlled by commercial publishers, i.e., where publishing is not the primary activity of the producing body”

vez analizado más en detalle el contenido de los mismos, se comprueba que solamente tratan *54 casos de implantación SAFe diferentes*, ya que varios artículos se centran en casos coincidentes.

A partir de esa información, los autores llevan a cabo un estudio para obtener información empírica acerca de los beneficios y retos de adoptar el framework SAFe, que tiene las siguientes conclusiones:

- *Principales beneficios*

Mejora en la transparencia de la información, mayor alineamiento entre equipos y con la organización, incremento de la productividad (más productos entregados), mayor predictibilidad del proceso y reducción en el *Time to Market*.

- *Retos más importantes*

Resistencia al cambio alta, dificultades en la organización de la primera reunión de *PI (Product Increment)* y percepción de regresar a prácticas de orientación predictiva.

Los autores analizan también si existe alguna diferencia de opinión entre los artículos obtenidos en la literatura gris y los publicados en fuentes académicas. En este sentido se encuentra una cierta diferencia en cuanto a los beneficios de negocio reportados. Mientras que en la literatura científica no se incluye información al respecto, en la literatura gris se resaltan beneficios como la reducción del *Time to Market*, la reducción de costes o la mejora de la satisfacción del cliente. En lo referente a los retos señalados, no se encuentra diferencia entre ambas fuentes de información.

Experiencias con LeSS

Teniendo en cuenta los resultados obtenidos para el caso de SAFe (escasa literatura académica encontrada que no aporta diferencias significativas respecto a la literatura gris), el estudio para el resto de *frameworks* se limita al análisis de los casos de estudio encontrados en los sitios web oficiales.

El número de casos encontrados para LeSS es menor que para SAFe, lo cual se justifica en base al menor grado de implantación de este *framework* en el mercado. Para el caso de LeSS, se han encontrado 27 referencias de casos de estudio publicadas, de las cuales se extraen las siguientes conclusiones:

- *Principales beneficios*

Mejora de la calidad y de la productividad, incremento en la transparencia del avance del trabajo y reducción del Time to Market.

- *Retos más importantes*

Resistencia al cambio, dificultad en el aprendizaje de las nuevas prácticas y complejidad de la puesta en marcha de prácticas DevOps (automatización de test y despliegue continuo)

Como puede observarse, los resultados obtenidos son muy similares a los obtenidos para el *framework* SAFe.

En los artículos, también se señalan los factores principales de éxito identificados en el proceso de implantación de la metodología. De entre ellos, pueden destacarse los siguientes como los más importantes:

- *Apoyo de la dirección.* El disponer del respaldo de los líderes de la organización y que tomen un papel activo en la transformación, es uno de los factores clave para conseguir el éxito del proceso.
- *Formación adecuada en todos los niveles de la organización.* La implantación de metodologías de escalado *Agile* supone cambios drásticos que impactan de forma profunda en la cultura de la empresa. La adecuada formación de todos los involucrados en el proceso es fundamental para lograr que dichos cambios sean entendidos de forma adecuada.
- *Cambio en la estructura organizacional de la empresa.* La organización del personal en base 'a funciones' no es eficiente. El objetivo principal es la optimización del tiempo de entrega del producto al cliente (flujo de valor) y, para ello, es necesario generar estructuras de trabajo multidisciplinarias que tengan contacto directo con el cliente y que minimicen dependencias.
- *Implantación gradual del proceso.* Todas las experiencias publicadas (excepto una, en la que el cambio se aborda de manera global en la organización), comienzan con la implantación de técnicas *Agile* a nivel de equipo (*Scrum*). Posteriormente se amplía el alcance de forma gradual, incluyendo progresivamente más equipos que se coordinan mediante prácticas propias de escalado.

Experiencias con DAD

Para el caso de DAD, el número de casos de estudio encontrados es incluso más reducido, limitándose a 10 experiencias encontradas en el sitio web de DAD. Los resultados que se extraen de la lectura de dichos artículos o presentaciones son muy similares a los señalados hasta el momento:

- Como *principales beneficios*, además de los comentados hasta el momento (mejor calidad, mayor productividad, alineamiento entre los distintos equipos), se señala de forma recurrente el incremento en la satisfacción de los empleados.
- En cuanto a los *principales retos* encontrados, en los artículos no se hace referencia explícita a esta cuestión. En cualquier caso, todos los casos publicados pasan por un proceso muy cuidadoso de formación y comunicación interna, de lo que puede deducirse que los principales retos encontrados se encuentran relacionados con el importante cambio de mentalidad necesaria y la dificultad en la adopción de estas nuevas prácticas.

Experiencias con otros *frameworks*

Para el resto de los *frameworks* presentados en este estudio (*scrum de scrums* y el método usado en *Spotify*), no se ha encontrado información relevante sobre implantaciones a gran escala, con lo que no pueden extraerse conclusiones acerca de los beneficios y retos asociados a los mismos.

3.2.3. Limitaciones en los estudios y conclusiones

Las limitaciones principales que pueden identificarse de los artículos analizados en este estudio del arte son las siguientes:

- *Escasa información disponible*, sobre todo en fuentes científicas. A pesar de que el tema genera un interés creciente, la información encontrada sobre experiencias de implantación de métodos de escalado *Agile* no es demasiado extensa. En las búsquedas realizadas en bases de datos académicas se encuentra diferente información sobre las dificultades existentes cuando se pretende aplicar *Agile* en entornos de gran escala, pero no sobre las soluciones finales adoptadas. Los estudios encontrados sobre dichas dificultades, suelen estar basados en

información recogida a través de encuestas a un conjunto de personas involucradas de alguna forma en el proceso. En general, se presentan estudios muy concretos con muestras no muy amplias, por lo que la mayoría de los autores coinciden en que se tratan de estudios preliminares que podrían usarse como punto de partida para futuras investigaciones pero no como información concluyente.

- *Información posiblemente sesgada.* La mayor parte de la información obtenida sobre implantaciones de métodos de escalado *Agile*, se ha recogido de los propios sitios web corporativos de SAFe, LeSS y DAD. Eso hace suponer que, posiblemente, la información presentada sea parcial y tienda a potenciar los beneficios y aspectos positivos de las implantaciones en vez de ser totalmente objetiva

A pesar de las limitaciones expuestas, la información encontrada al margen de la propia de los *frameworks* de escalado, aunque escasa, corrobora los datos presentados, lo cual le confiere cierta fiabilidad. Además, aunque la información pudiera tener cierto sesgo, muestran las opiniones de expertos de prestigio en *Agile* y profesionales de importantes empresas que se han enfrentado a procesos muy complejos de transformación *Agile* a gran escala, con lo que se considera que merece la pena tenerlas en cuenta.

En cualquier caso, es cierto que la información encontrada es interesante pero insuficiente para poder obtener resultados concluyentes. Sería necesario disponer de una muestra mucho más amplia para tener evidencias suficientes como para confirmar los beneficios reales de cada *framework* de escalado. La adopción de prácticas *Agile* y todo el conjunto de metodologías y *frameworks* relacionados con esta tendencia, está en auge. No solo en el ámbito empresarial sino también en entornos académicos comienza a despertar interés y a estudiarse en más profundidad. En vista de este éxito es de esperar que, en los próximos años, el número de implantaciones crezca y con ellas la información disponible en diversos ámbitos.

Por el momento, y para la elaboración de este trabajo, se considera que la información obtenida confirma los beneficios de considerar un *framework* de escalado para la implantación de *Agile* en grandes entornos de trabajo software y, de entre ellos, seleccionar SAFe para un análisis en mayor profundidad por ser, actualmente, el de uso más extendido.

3.3. Scaled Agile Framework (SAFe)

3.3.1. Mentalidad y principios SAFe

SAFe, como se ha comentado anteriormente, es el marco de trabajo más extendido de escalado *Agile*. Se plantea en este apartado un estudio más detallado de este *framework*, lo que permitirá la identificación de sus fortalezas y debilidades de cara al planteamiento de una metodología propia de aplicación a empresas de desarrollo software con equipos distribuidos (23,26).

SAFe, está basado en filosofías *Lean* y *Agile*, cuyos conceptos fundamentales son los siguientes:

- *Lean* deriva del sistema de producción de Toyota desarrollado entre las décadas de 1950 a 1980. Es un método de producción que, en contraposición al sistema de producción en masa (que demanda una gran cantidad de recursos e inversión), propone un sistema que se ajuste a la demanda y optimice el uso de recursos, mejorando la calidad y eliminando las tareas que no aportan valor.
- *Agile* surge de la industria del software y se centra en la flexibilidad y *adaptación al cambio*. Su base, por lo tanto, es el desarrollo iterativo orientado a la entrega temprana de valor y así poder reorientar los desarrollos. *Agile* además pone a 'las personas' por encima de cualquier 'procedimiento o herramienta'.

SAFe postula nueve principios (basados en las filosofías anteriores) y que son la base de las prácticas que propone. De los nueve principios comentados, se destacan los dos siguientes por su trascendencia.

- *Aplicar el pensamiento sistémico (System Thinking)*. Un sistema está formado por los subsistemas que lo componen y las relaciones entre los mismos (personas, procesos, herramientas, etc.). El optimizar un solo componente no optimiza el sistema y un sistema no evoluciona más rápido que su componente más lento, con lo que la visión holística del sistema es fundamental.
- *Descentralizar la toma de decisiones*. Este principio es fundamental para lograr agilidad, pero para que sea eficaz, es necesario que la gerencia de la empresa defina y transmita claramente la estrategia.

Derivada de esta filosofía, SAFe aporta una visión radicalmente distinta a la que promulgan las metodologías de tipo 'predictivo' (por ejemplo, PMI-PMBok).

Uno de los conceptos que introduce SAFe y que puede resultar más disruptivo respecto a PMI-PMBok es la diferencia en cuanto al ‘alcance’ a gestionar. Mientras que en PMI-PMBok el ‘proyecto’ es el centro de la metodología, en SAFe no existe ese término. Es más, SAFe prescinde intencionadamente de su uso y, de esta forma, evita las connotaciones que las metodologías tradicionales han impreso a la palabra ‘proyecto’. Según PMI-PMBok “*un proyecto es un esfuerzo **temporal** que se lleva a cabo para crear un producto, servicio o resultado único* (1)”. Esta naturaleza temporal implica que, una vez finalizado el proyecto, se da por concluido el trabajo y se disuelven los equipos que serán reubicados en otro proyecto. SAFe, al contrario, da una continuidad al alcance (que materializa en lo que denomina ‘flujos de valor’) y a los equipos de trabajo, que se encargan de aprender y evolucionar dichos ‘flujos de valor’ de forma continua.

A partir de esta filosofía y principios, SAFe propone una serie de artefactos (roles, ceremonias, herramientas, etc.) que ayudan a *desarrollar sistemas minimizando el tiempo de entrega y aportando el mayor valor y calidad posibles*.

3.3.2. Configuraciones SAFe

El *framework* SAFe ofrece diferentes configuraciones orientadas a soportar las necesidades en diferentes contextos (desde organizaciones pequeñas con varios equipos de trabajo, hasta entornos complejos en los que colaboran miles de personas). SAFe ofrece mecanismos de coordinación a diferentes niveles:

- *A nivel de equipo*. Los diferentes equipos trabajan con una cadencia común para definir, desarrollar y testear historias del *backlog*.
- *A nivel de programa*. Diferentes equipos de trabajo se sincronizan para desarrollar sistemas más complejos.
- *A nivel de portfolio*. Este es el nivel superior de SAFe que permite engranar el desarrollo *Agile* (implementado en los niveles de equipo/programa) con los objetivos estratégicos.

A continuación, se describen los artefactos que es preciso incorporar en los procedimientos de gestión de proyectos para cubrir los niveles de equipo, programa y portfolio:

Nivel de equipo

SAFe, a nivel de equipo, recomienda la incorporación de ‘prácticas *Scrum*’. Dichas prácticas son las que ya se han comentado al comienzo de este trabajo: se trabaja en pequeños equipos multidisciplinares en base a iteraciones que permiten entregar valor de forma continua mediante el desarrollo de un Backlog mantenido y priorizado por el *Product Owner* (cliente).

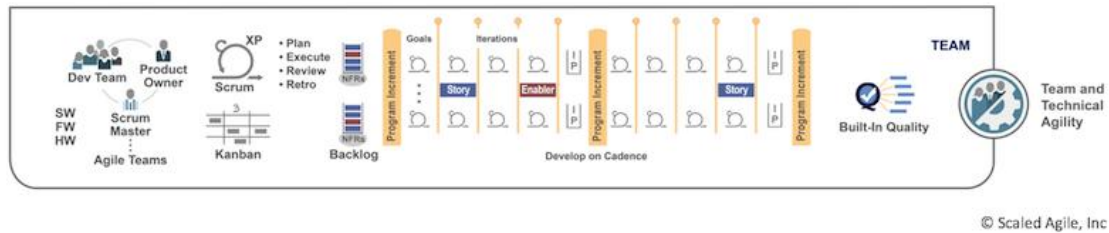


Figura 13. Representación del nivel de equipo en SAFe (23)

Este es el nivel inicial. Su operativa básica ha sido ya ampliamente descrita y se considera que no es necesario disponer de un mayor detalle para desarrollar el trabajo que nos ocupa.

Nivel de programa

El nivel de programa es fundamental en SAFe y permite implementar los mecanismos de coordinación a nivel operativo entre equipos de trabajo (equipos *Scrum*). El artefacto principal de SAFe es el ART (*Agile Release Train*). El ART permite alinear a la dirección, a los principales interesados y a los equipos de trabajo en torno a unos objetivos comunes. Los ART suelen estar formados por entre 5 y 12 equipos *Agile* (entre 50-125 personas) que operan según los siguientes principios:

- Las iniciativas que los ART desarrollan, se recogen en un *backlog* de programa. Dichas iniciativas, denominadas ‘*feature* o *enabler*’ (dependiendo de si su orientación es de negocio o técnica), provienen de un *backlog* de portfolio (*Épicas*), que se genera en un nivel superior y que se encuentra alineado con la estrategia empresarial.
- Los equipos trabajan en ciclos cortos de desarrollo (ciclos *Scrum*). De esta forma se genera un incremento de sistema potencialmente desplegable cada dos semanas.

- La coordinación de los equipos se articula a través de los *PI (Product Increment)*. Un PI es el conjunto de funcionalidades que serán desarrolladas en un intervalo concreto de tiempo. Es el equivalente a las *iteraciones/sprints* en *Scrum* pero a nivel de programa. La duración del PI es fija y suele ser entre 8 y 12 semanas.
- Los objetivos del PI se analizan y planifican en una reunión a la que deben asistir todos los miembros del *Tren de desarrollo (ART)*. En dicha reunión (similar al *sprint planning meeting*) se discuten las *Épicas* de mayor prioridad, se identifican dependencias y se planifica la entrega que va a generar el ciclo (entrega comprometida por el ART). Las *Épicas* se desdoblan en entidades menores que serán incluidas en las distintas iteraciones (*feature/enabler*).
- Para posibilitar los procesos de mejora continua y adaptación, en cada PI se reserva tiempo para realizar ajustes de los procesos, formación, investigación, validación y verificación de la solución completa con el cliente y preparación del siguiente PI.

En este nivel también se generan responsabilidades nuevas que son cubiertas por nuevos roles:

- *Release Train Engineer (RTE)*. Es el facilitador del proceso. Tiene unas funciones similares al *Scrum Master* pero a nivel ART: ayuda a eliminar impedimentos, gestiona dependencias, riesgos y facilita la mejora continua.
- *Product Manager*. Es el responsable del *backlog* de programa. Trabaja directamente con el cliente y con los *Product Owner* de los equipos *Scrum* de forma que se asegura que el trabajo de todos los equipos está alineado entre sí y con las necesidades del cliente. Además, es responsable de la priorización del *backlog* y de aportar la visión general a nivel de programa y de *roadmap*.
- *System Architect/Engineer*. Es una persona o equipo encargado de definir y comunicar la arquitectura global del sistema de forma que se asegure que los desarrollos están alineados entre sí y con el objetivo común. Definen los requisitos técnicos no funcionales (NFRs) y ponen en marcha lo que SAFe denomina como *Architectural Runway* que permite balancear la implementación de NFRs con la de *Épicas* de negocio.

- *Business Owners*. Son los miembros de dirección, responsables de la financiación de los proyectos y responsables finales de los resultados.

Para la priorización de los elementos en el *backlog* (bien sea de equipo, programa o portfolio), SAFe recomienda usar el criterio *WSJF* (*Weighted Shortest Job First*), es decir, acometer primero el trabajo cuya duración ponderada sea menor. El *WSJF* se define de la siguiente forma:

$$WSJF = \frac{\text{Coste del retraso (CoD)}}{\text{Duración de la tarea (JS)}} \quad (1)$$

Donde:

- *Coste del retraso/Cost of Delay (CoD)*. Viene determinado por la siguiente ecuación:

$$\begin{aligned} \text{Cost of Delay (CoD)} = \\ \text{User Business Value} + \text{Time Critically} + \\ \text{Risk reduction or Opportunity Enablement} \end{aligned} \quad (2)$$

User Business Value = Valor que la funcionalidad aporta al negocio.

Time Critically = Valoración de la criticidad asociada a los retrasos en la entrega de ese valor.

Risk reduction = Indicador que refleja si la tarea aporta además otros beneficios como una reducción del riesgo y/o una posibilidad de nuevas oportunidades

- *Duración de la tarea/Job Size (JS)* = Estimación de la duración o del tamaño de la tarea.

Para obtener el *WSJF*, no es preciso disponer de valores concretos de sus términos. En este caso lo importante es contar con estimaciones aproximadas que nos permitan comparar diferentes *Épicas* (cuanto más de compleja es una *Épica* que otra, si una aporta más valor que otra, etc.). Se suele usar los primeros valores de la ‘Serie de Fibonacci’ (1,2,3,5,8,13,21) para realizar esta valoración. El cálculo del *WSJF* permite ordenar el *backlog* según un criterio orientado a maximizar el flujo de generación de valor.

En la siguiente imagen se muestra el flujo del proceso que define SAFe a nivel de programa.

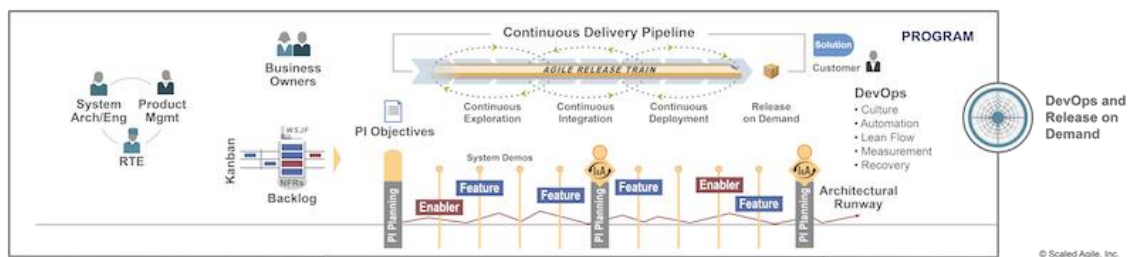


Figura 14. Representación del nivel de programa en SAFe (23)

Nivel de portfolio

Si bien el nivel de programa es el que orquesta los mecanismos de coordinación a nivel operacional, el *nivel de Portfolio* es el que realmente permite el escalado de la *agilidad* a nivel corporativo. En este nivel, el objetivo es extender la aplicación de los principios SAFe a la estrategia y los órganos de gobierno de la compañía. Se revisan los criterios de selección y financiación de las iniciativas, aligerando los procesos de forma que todas las estructuras de la organización se orienten hacia una entrega ágil de valor. Para ello es preciso cambiar la mentalidad y pasar de una optimización en la utilización de los recursos a una optimización en el tiempo de entrega del producto.

El proceso para la gestión del Portfolio que promueve SAFe es bastante complejo y maneja conceptos como la definición de la estrategia, la financiación de la inversión y la gestión de los presupuestos entre otros. Aunque el análisis de estos temas queda fuera del alcance de este trabajo, sí se incluirá la parte más operativa del proceso que trata de cómo alinear los conceptos anteriores con los niveles de programa y equipo. Dicha alineación se realiza a través del flujo de selección y priorización de las iniciativas a desarrollar por dichos niveles inferiores. El incluir esta parte del proceso correspondiente al ámbito del portfolio permite tener una perspectiva global de cómo se desarrolla el ciclo completo de gestión de proyectos.

El elemento fundamental en SAFe a nivel de portfolio es la *Épica*. El concepto de *Épica* en SAFe es diferente al utilizado en *Scrum*. En este caso, una *Épica* es una iniciativa suficientemente grande que debe ser analizada y aprobada previamente a su desarrollo. El análisis de la *Épica* se materializa en un documento denominado *Lean Business Case* que incluye información tanto funcional como financiera: se definen los objetivos de la *Épica*, los resultados

que se prevé obtener, los costes y plazos estimados para el desarrollo, etc. Además, también se define el *MVP (Minimum Viable Product)*, que es la funcionalidad mínima que es necesario implementar para que esa *Épica* pueda entregarse al cliente. *El MVP* se despliega y, en función de los resultados, se continúa con los desarrollos o se da por finalizado el trabajo. Las *Épicas* aprobadas se integran en un tablero Kanban de Portfolio en el que se refleja el flujo del proceso.

Los roles que trabajan a nivel de portfolio, son los siguientes:

- *Epic Owner/Propietario de la Épica*. Es una figura principal en la gestión del Portfolio y es el responsable de coordinar y dar seguimiento a las *Épicas*. Además, define el MVP (mínimo producto viable) y el caso de negocio. Este rol es, en resumen, es similar al promotor de un proyecto.
- *Enterprise Architect/Arquitecto Empresarial*. Se encarga de lanzar y coordinar las *Épicas* de tipo 'enabler' (requisitos no funcionales/NFRs) de forma que exista un balance apropiado entre el desarrollo de funcionalidades de negocio y técnicas. También proporciona directrices técnicas que permiten una homogeneidad en cuanto al modo de diseño y codificación, lo cual facilita la reutilización de componentes.
- *Lean Portfolio Management/Gestión del Portfolio Lean (LPM)*. Es el nivel más alto en la jerarquía de toma de decisiones en cuanto a Portfolio. También asume el rol principal en el ámbito financiero.

Para finalizar esta parte, y a modo de resumen, en la siguiente imagen se incluyen los diferentes roles y actividades que SAFe define y que permiten coordinar el trabajo en los niveles anteriormente descritos.

	Portfolio	Programa	Equipo
Roles	Lean Portfolio Management Epic Owner Enterprise Architect	Release train engineer System architect/engineer Product manager Business Owner	Product Owner Scrum master Scrum team
Activities	Define value streams Define portfolio backlog	Form and launch ARTs PI Plannig - Define program backlog ART Sync (scrum of scrums, PO sync) System demo Inspect and Adapt (I&A)	Sprint planning Sprint execution System demo Iteration retrospective

Tabla 2. Roles y artefactos SAFe.

3.4. Técnicas y herramientas Agile

Las técnicas y herramientas que soportan la ejecución de las actividades involucradas en un proyecto son elementos fundamentales dentro de una metodología de gestión de proyectos. Las necesidades que se plantean en entornos *Agile*, son diferentes a las existentes en entornos de orientación predictiva y, por lo tanto, las herramientas utilizadas son diferentes (7,17,23). En *Agile* no encontramos diagramas de Gant para realizar la planificación del proyecto y los procesos de estimación del coste y duración de las tareas son más ligeros. En este apartado se hará una revisión de algunas de las técnicas y herramientas de uso más extendido en estos entornos.

Tableros Kanban

Una de las principales herramientas que se utilizan en los modelos de gestión *Agile*, son los **tableros Kanban**. Dicha herramienta, fue desarrollada por Taiichi Ohno como una forma de controlar la producción en Toyota. Este método persigue ajustar la producción a la demanda lo más posible, controlando el flujo del proceso de tal forma que cada fase se ejecute justo en el momento necesario (just in time): ni ‘antes’ ni ‘después’. Realizar la tarea ‘antes’ de tiempo provoca sobrecostes (almacenaje innecesario) y ejecutarla ‘después’ genera retrasos. Para controlar ese flujo de trabajo, se creó el *tablero Kanban* formado por un panel dividido en columnas que representan las fases del proceso. Sobre ese panel se colocan tarjetas que representan las diferentes tareas a realizar de forma que, el avance, se representa de forma visual.



Figura 15. Representación de tablero Kanban con seis estados
Fuente: <https://kanbanize.com>

Para permitir que el flujo de trabajo avance de forma continua es necesario limitar el WIP (Work in Progress). Cada fase del proceso tiene un número máximo de tareas que pueden ser ejecutadas en paralelo: el hecho de avanzar más en unas partes del proceso que en otras, genera cuellos de botella que son fácilmente identificables en el panel Kanban.

Otra ventaja importante de este panel es que permite hacer visible a la organización el estado de las tareas y su avance, lo cual aporta transparencia y ahorra tiempo de reuniones.

Tableros de programa

El Tablero de programa permite visualizar el trabajo planificado para el *PI* (*Product Increment*) y sus dependencias. Este tablero se obtiene como resultado de la reunión de Planificación del *PI* y se va actualizando a medida que avanza el trabajo.

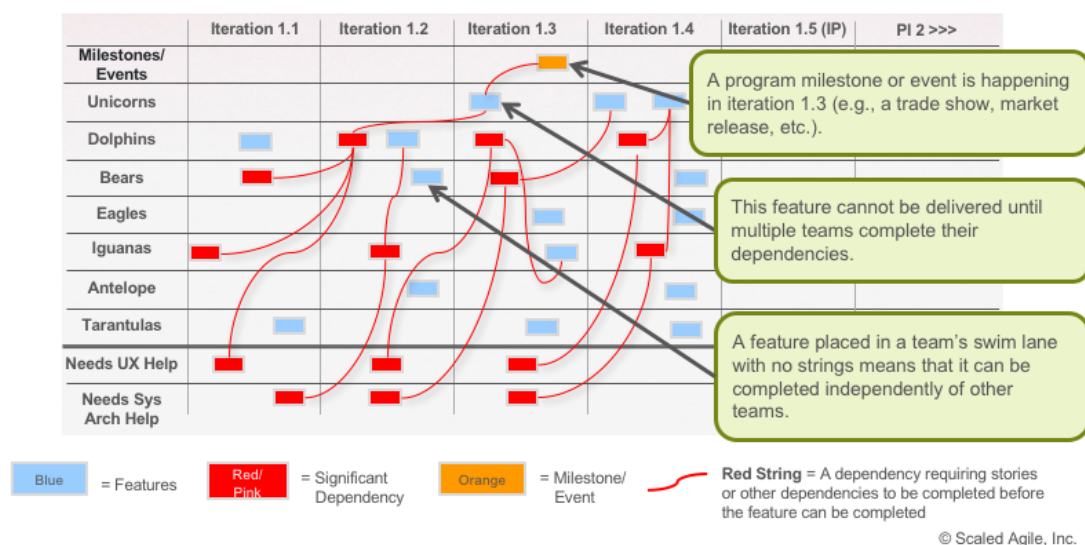


Figura 16. Representación de un tablero de programa (23)

MoSCoW

Es una técnica orientada a la priorización de requisitos en base a su clasificación en cuatro categorías:

Must: Características obligatorias.

Should: No obligatorias pero altamente recomendables.

Could: Elementos adicionales (deseables pero no imprescindibles)

Wont: Funcionalidades que es mejor evitar

WSJF (Weighted Shortest Job First)

Del mismo modo que MoSCoW, el WSJF es una técnica orientada a la priorización de requisitos. Esta métrica se ha detallado en el apartado 3.3.2 *Configuraciones SAFe* de este documento y se calcula teniendo en cuenta el ‘coste del retraso’ respecto al ‘tamaño de la tarea’. Priorizando las tareas de menor WSJF, se acometen primero aquellos elementos más cortos y que más valor aportan (produce el mayor valor en el menor plazo posible).

Modelo de negocio Canvas

Aunque este trabajo se orienta más a la parte operativa de la gestión de proyectos que a la estratégica, se señala el *modelo ‘Canvas’* como una forma ‘ligera y ágil’ de diseñar un plan de negocio. Ésta es una herramienta gráfica que permite plasmar el modelo de negocio de una forma sencilla. A través de la reflexión estratégica de las 9 secciones que propone este modelo, se identifican los aspectos clave a tener en cuenta a la hora de lanzar un producto al mercado.

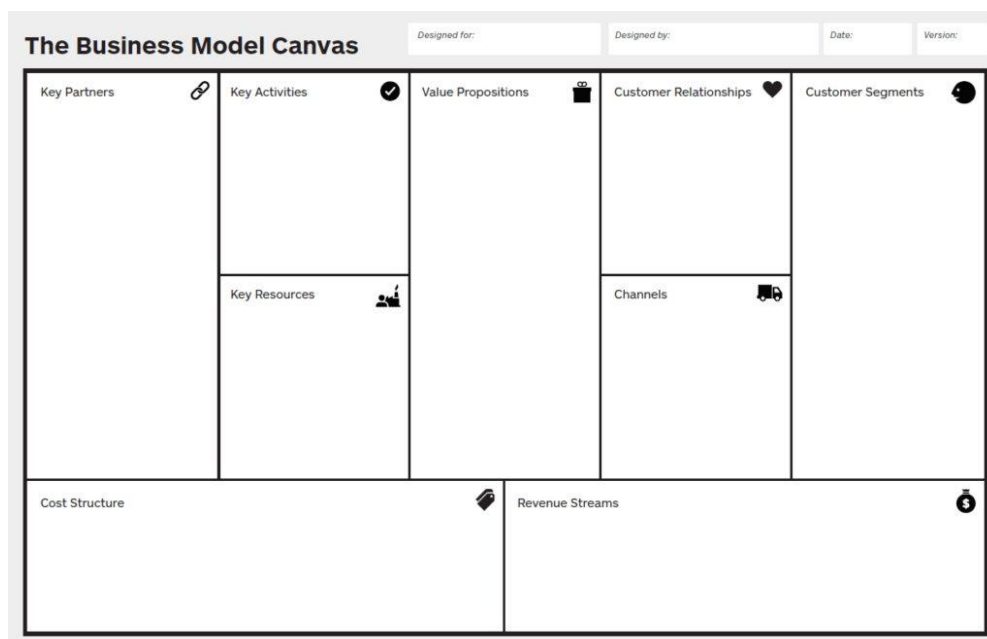


Figura 17. Representación de un Canvas de negocio

Fuente: <https://modelocanvas.info>

Herramientas para el desarrollo software

Las metodologías Ágiles persiguen la satisfacción del cliente mediante la entrega temprana y continua de software. Para poder conseguir esa entrega

continua es preciso desarrollar mecanismos que permitan automatizar los procesos entre los equipos de Desarrollo y de Operaciones de forma que se pueda compilar, probar y publicar software con rapidez y fiabilidad (CI-Continuous Integration/CD-Continuous Deployment). Al movimiento que impulsa esa colaboración entre los equipos de Desarrollo y Operaciones para permitir esta agilidad, se denomina ‘DevOps’. No es objeto de este documento realizar un estudio de las herramientas que actualmente existen para implementar mecanismos de integración y despliegue continuo. Sin embargo, es importante señalar que el poner en marcha estos mecanismos es fundamental para lograr la ‘*agilidad* en la entrega’, que es lo que realmente aporta valor.

Herramientas software colaborativas y de gestión de proyectos

El uso de herramientas software colaborativas y de gestión de proyectos resulta imprescindible en entornos distribuidos. En este sentido las necesidades que es preciso cubrir son las siguientes:

- Creación, priorización y mantenimiento de *backlogs* a diferentes niveles (equipo, programa y portfolio)
- Organización y visualización del flujo de trabajo mediante tableros Kanban
- Comunicación y colaboración entre los equipos (wiki, chat, reuniones en línea, videoconferencia, acceso a documentación compartida, etc.)
- Configuración de métricas y herramientas orientadas a la toma de decisión y mejora de los procesos ágiles a diferentes niveles (equipo, programa y portfolio)

3.5. Métricas Agile

Según el Manifiesto *Agile* (3) la métrica fundamental que informa del progreso en *Agile*, es el ‘software funcionando’. En *Agile*, el avance del trabajo ‘se puede percibir directamente’ (tableros kanban, feedback de clientes, etc.), con lo que no sería necesario aportar más información. No obstante, las métricas son importantes a nivel de negocio y deben fluir del nivel de equipo hacia los niveles superiores. A continuación se presentan un conjunto de métricas que pueden aplicarse en entornos de escalado *Agile* (7,17,23).

Diagrama de flujo acumulado

Este diagrama, relacionado directamente con la herramienta Kanban, representa la forma en el que las 'historias' avanzan a lo largo del tablero. Es una herramienta muy potente que permite visualizar el avance del proceso y encontrar puntos de mejora que permitan optimizar el flujo.

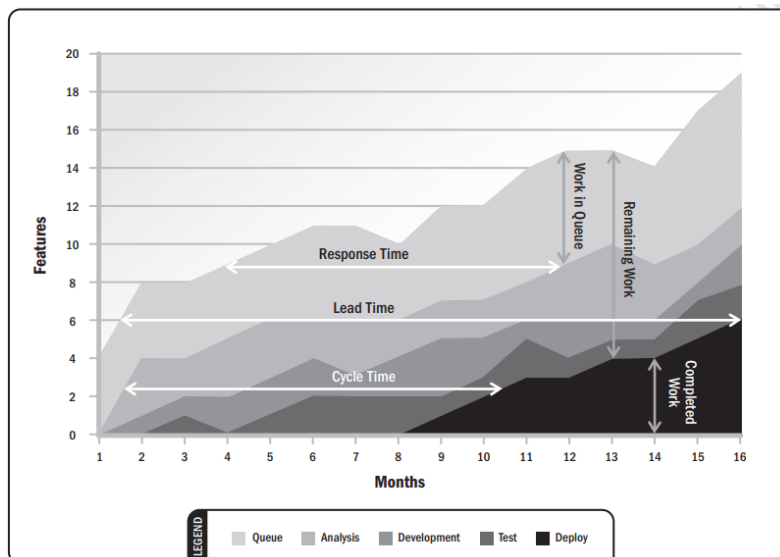


Figura 18. Diagrama de flujo acumulado para kanban de 5 estados (17)

Diagrama Burn-up de Coste-Valor

Es una gráfica muy interesante a aplicar cuando se trabaja con orientación SAlFe. En dicha gráfica se representan dos curvas:

- Una correspondiente al avance del coste del proyecto a medida que avanzan las Épicas.
- Otra correspondiente a cómo avanza el valor aportado al proyecto a medida que avanzan las Épicas

En entornos *Agile* lo habitual es que *el valor* al comienzo del proyecto sea muy alto pero, llegará un momento, a partir del cual el valor aportado ya no es tan importante (pero el coste sigue incrementándose). En ese punto es donde es preciso considerar si continuar con el proyecto o darlo por finalizado.

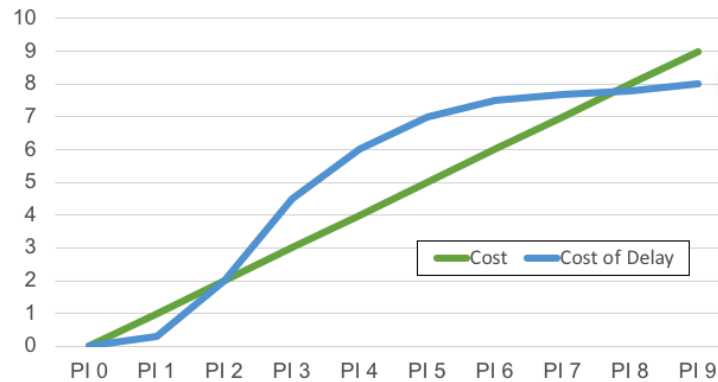


Figura 19. Diagrama Burn-up de Coste-Valor
Fuente: <https://www.ppmschool.com>

Informe de progreso de las Épicas / Epic progress report

Es una métrica similar a la de proyectos más tradicionales y permite hacer un seguimiento a nivel de *Épica*. En terminología SAFe, las *Épicas* constituyen una ‘idea’ a nivel de negocio. Dichas ‘ideas’ tienen envergadura suficiente como para plantear su viabilidad económica, realizar un análisis de la solución previa a la implementación y plantear un MVP (Minimum Viable Product).

En la siguiente imagen se muestra un diagrama correspondiente a esta métrica donde pueden verse representados los valores correspondientes a las estimaciones realizadas para el desarrollo del trabajo (planificado y actual).

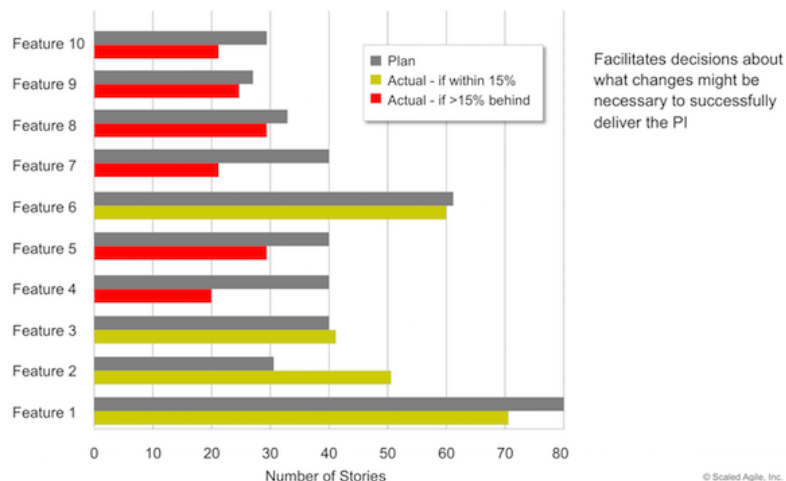


Figura 20. Diagrama de progreso de las Épicas (23)

Valor ganado Agile

La técnica del valor ganado (EVM – Earned Value Management) es una de las principales herramientas utilizadas en entornos PMI-PMBok predictivos. El concepto del Valor Ganado, en principio, no parece aplicable a *Agile*, ya que

representa la desviación de los resultados obtenidos respecto a los planificados. A pesar de que, en *Agile* no se dispone de una planificación ni de alcance cerrado para la solución completa, los equipos comprometen un alcance a nivel de Iteración o de *PI (Product Increment)*. La representación gráfica del avance de los resultados obtenidos respecto a los planificados a nivel de *PI (Product Increment)*, ofrece información interesante tanto para los equipos de desarrollo (permite corroborar la velocidad del PI) como para la dirección (obtiene información resumida del avance de los desarrollos).

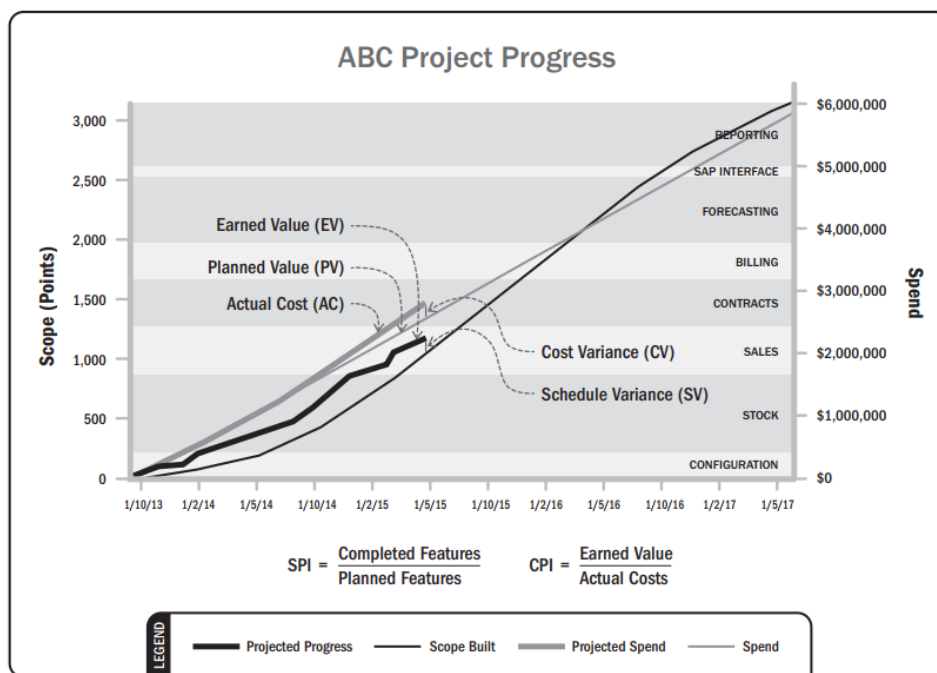


Figura 21. Valor ganado Agile (17)

4. PROPUESTA DE METODOLOGÍA HÍBRIDA

4.1. Objetivos y justificación de la metodología

Una vez finalizado el estado del arte presentado en el apartado anterior, es preciso retomar la pregunta planteada al comienzo de este trabajo que cuestiona la validez de las metodologías *Agile* en entornos de desarrollo software con equipos distribuidos. En el estado del arte se amplía dicha cuestión analizando tanto la validez de las metodologías *Agile* en estos entornos, como la de las metodologías tradicionales, incluyendo en el estudio las ventajas e inconvenientes de ambas aproximaciones.

Para hacer una propuesta de las metodologías más apropiadas a utilizar en grandes empresas dedicadas al desarrollo software, es preciso tener en cuenta que el entorno en el que trabajan es altamente competitivo y requiere de gran rapidez y flexibilidad ante cambios. En este escenario, las metodologías tradicionales de gestión de proyectos, que eran válidas hace 40 años cuando la velocidad de innovación era menor, precisan unos tiempos de respuesta y presentan una rigidez que son incompatibles con las necesidades del sector. Es evidente, por lo tanto, que la aplicación de procedimientos tradicionales no cubre las necesidades planteadas y es preciso adoptar otro tipo de solución.

Por otra parte, las metodologías *Agile* plantean formas de trabajo orientadas a optimizar los tiempos de respuesta permitiendo además una alta flexibilidad. En el estado del arte elaborado en este documento se ha realizado un estudio de estas metodologías señalando las limitaciones y problemas que presentan cuando se utilizan en grandes sistemas software desarrollados por equipos distribuidos. Las dificultades de comunicación, la gestión de las dependencias y la deuda técnica generada, son los principales problemas que se encuentran al emplear '*Agile* en estado puro' en grandes entornos. Con el objetivo de dar respuesta a esos problemas han surgido un conjunto de *frameworks* que se agrupan bajo el nombre de '*Escalado Agile*'. Dichos *frameworks* se construyen sobre una base *Agile* a la que incorporan distintos artefactos (roles, ceremonias, herramientas) que permiten establecer mecanismos adicionales que cubren las necesidades especiales planteadas por los grandes sistemas.

Teniendo en cuenta lo anterior, podría suponerse que la aplicación de un *framework* de escalado *Agile*, como por ejemplo SAFe, debería ser suficiente

para resolver el problema planteado. Sin embargo, tras un tiempo de experimentación con este tipo de *frameworks*, afloran ciertas deficiencias que pueden poner en riesgo el éxito de los proyectos. La base *Agile* sobre la que se construyen aporta muchos beneficios pero, como contrapartida, también presenta ciertos inconvenientes derivados del escaso nivel de concreción en cuanto a las responsabilidades a asumir o las actividades a realizar. En realidad, esa falta de concreción es intencionada en *Agile* y persigue potenciar la descentralización de algunas actividades e, incluso, de la toma de decisiones. En grandes instituciones, acostumbradas a la rigidez de los métodos tradicionales de gestión de proyectos, esto puede suponer un problema generando, en ocasiones, vacíos de responsabilidad que no son cubiertos fácilmente de forma 'autónoma'.

Por el contrario, las metodologías predictivas, en concreto PMI-PMBok, tienen un altísimo grado de concreción en sus procedimientos y, su uso permite asegurar que todas las actividades necesarias para la gestión de un proyecto han sido consideradas y tienen un responsable. PMBoK, además, propone un amplio conjunto de técnicas y herramientas adecuadas para cada caso.

Por los motivos anteriores, en este trabajo se plantea como solución el desarrollo de un conjunto de procedimientos que, sobre una base de escalado *Agile*, incorporen algunos elementos propios de metodologías predictivas. De esa forma, se añade un pequeño nivel de concreción y control en aquellas fases del proyecto más críticas o que, por algún otro motivo, se considera necesario reforzar. La identificación de los aspectos a reforzar se realiza en base al estudio previo de metodologías, a la tipología de proyectos en los que se prevé aplicar esta metodología (grandes proyectos desarrollados por equipos distribuidos) y a la experiencia personal.

4.2. Selección de frameworks para la gestión del portfolio

Teniendo en cuenta la información presentada hasta el momento, se seleccionan los siguientes *frameworks* a partir de los cuales se elaborará un conjunto de procedimientos de aplicación en el entorno definido:

- **SAFe**, como *framework* híbrido orientado a escalado *Agile*.

- **PMI-PMBok**, como metodología complementaria para cubrir aquellas áreas de conocimiento que SAFe no contempla de forma directa o que se precisa resolver de manera más formal.

Esta selección se considera como una opción válida para resolver el problema planteado por los siguientes motivos:

- SAFe es el *framework* de escalado *Agile* de uso más extendido.
- SAFe implementa *Scrum* a nivel de equipo. En la actualidad, esta metodología *Agile* es una de las de uso más extendido en este nivel y sus resultados han sido ampliamente probados.
- SAFe contempla, dentro de su framework, diferentes niveles que incluyen la gestión del ciclo completo de producto: desde la concepción de la idea hasta el desarrollo y la evaluación final del resultado.
- SAFe prevé un tiempo reservado para la innovación. Las empresas de desarrollo software deben dedicar tiempo y recursos a evolucionar técnicamente, lo cual favorece la mejora continua.
- PMI-PMBok es la metodología de tipo predictivo más ampliamente usada y contempla dentro de sus procesos todas las actividades que deben ser consideradas en la gestión de un proyecto complejo. Además, también incorpora recomendaciones y prácticas concretas que pueden ser de utilidad bajo cualquier entorno de trabajo, tanto predictivo como *Agile*.
- Tanto SAFe como PMI-PMBok son marcos con un alto nivel de concreción en sus prácticas y control sobre el proceso, lo cual facilita la implantación en aquellos casos en el que las empresas no se encuentran maduras en el campo de la gestión de proyectos.

Además, SAFe y PMI-PMBok consideran dentro de sus prácticas, no solo la gestión de proyectos, sino también la gestión de los niveles de programa y portfolio.

- En el *nivel de programa* se gestionan distintas partes de un proyecto o distintos proyectos relacionados.
- En el *nivel de portfolio* se gestionan las distintas ideas o iniciativas en marcha en una compañía con o sin relación entre sí.

La elaboración de una *metodología que contemple el nivel portfolio* permite regular las relaciones que se generan en la compañía cuando se gestionan

varios proyectos en paralelo (relacionados o no), lo que es el escenario habitual.

4.3. Hibridación PMI-PMBok / SAFe

A continuación, se plantea cómo pueden complementarse las metodologías PMI-PMBok y SAFe de forma que se logren los objetivos propuestos.

Teniendo en cuenta la tipología de proyectos objeto de este estudio (grandes proyectos software con equipos distribuidos), se plantea basar los procedimientos en una metodología con una base SAFe, ya que se considera fundamental mantener los beneficios que aporta *Agile* en cuanto a agilidad y flexibilidad. SAFe resulta un buen punto de partida, pero presenta debilidades que son necesarias resolver. PMI-PMBok, por su parte, ofrece un marco muy completo de trabajo donde están identificadas y controladas todas las actividades que pueden intervenir en el desarrollo de un proyecto. Además, en la actualidad, el uso de PMI-PMBok está muy extendido y, por lo tanto, los procesos y áreas de conocimiento de PMI-PMBok son conocidos en aquellas empresas con experiencia en gestión de proyectos.

Teniendo esto en cuenta, se realiza a continuación un estudio que permite establecer un paralelismo entre las prácticas SAFe y cada una de las diez Áreas de Conocimiento de PMI-PMBok. El objetivo es encontrar aquellas cuyo alcance sea más difícilmente identificable dentro del marco SAFe y valorar si es preciso reforzar algún aspecto concreto de cara a la definición del procedimiento de gestión de proyectos.

Gestión de la integración

SAFe es básicamente ‘integración’. Su objetivo es la coordinación de todas las partes para lograr *‘desarrollar el máximo valor de negocio con el tiempo de entrega sostenible más corto y con el mayor nivel de calidad posible’*.

En lo referente a esta área de conocimiento, se toma como referencia los procedimientos SAFe.

Gestión del alcance

Las *Épicas, features, enablers, historias*, etc., que se definen a lo largo del proceso, constituyen el alcance del proyecto.

Aunque la filosofía del ‘alcance’ es diferente en ambas aproximaciones, se considera que, en este caso, no es necesario aportar artefactos adicionales a los anteriores.

Gestión del tiempo

La planificación de los *PI (Product Increment)*, la gestión de los Kanban de portfolio, programa y equipo, etc.; son tareas relacionadas con la gestión del tiempo. Para planificaciones a más largo plazo, SAFe también contempla la elaboración de *roadmaps* a nivel de *PI (Product Increment)* o Solución.

De la misma forma que el alcance, la concepción del cronograma en ambos casos es muy diferente. No obstante, no se considera necesario reforzar este punto ya que se trata de un pilar básico de *Agile/SAFe*.

Gestión del coste

La elaboración del *Lean Business Case* y la continua priorización en base al criterio WSJF (*Weighted Shortest Job First*), pueden considerarse dentro de esta área de conocimiento. Los anteriores son mecanismos orientados a la aprobación y priorización, sin embargo, SAFe no propone herramientas para el control de costes a lo largo de la ejecución de un proyecto. En realidad, SAFe rechaza de manera tajante el control de costes al estilo tradicional. Dicho control, según este *framework*, limita la innovación y ralentiza la entrega de valor. SAFe se encarga de financiar los ‘flujos de trabajo’ que tienen un coste fijo y no entra en más detalle (el objetivo es, que el equipo genere valor demostrable y se confía plenamente en que los equipos están aprovechando los recursos de forma óptima para obtener esa generación de valor).

Esta propuesta de SAFe es controvertida y complicada de introducir en empresas que se encuentran en fases iniciales de implantación de *Agile*. Es por ello que, en la propuesta de hibridación que se plantea en este documento, se sugiere realizar un seguimiento de la evolución del *PI (Product Increment)* mediante una adaptación de la técnica del ‘valor ganado’ (valor ganado *Agile*).

Gestión de la calidad

SAFe está basado en filosofía *Lean* y como tal, la calidad es una de sus prioridades. Además, cualquier metodología de naturaleza iterativa se basa de

alguna forma en el ciclo PDCA de Deming (mejora continua). SAFe también incorpora mecanismos que previenen la falta de calidad derivada de la descoordinación entre equipos *Agile* mediante la incorporación de la figura del 'Arquitecto' (Sistema o Empresa). Además, se incluye dentro del proceso el concepto de *Architectural Runway* y un tiempo dedicado a la investigación y retrospectiva, lo que redundará en un proceso de mejora continua.

Por su parte, PMBoK define la calidad como el 'grado de cumplimiento de los requisitos', concepto no compatible con la filosofía *Agile*. En ese sentido, cuando hablamos de calidad en SAFe (para desarrollo software) podemos identificarla con la satisfacción del cliente y con métricas acerca de los defectos en el software.

Gestión de los recursos del proyecto.

Si se consideran los recursos humanos del proyecto, este es uno de los procesos de filosofía más diferente en ambos *frameworks*. En SAFe no tiene sentido la gestión de los recursos humanos que promulga PMI-PMBoK. No existe ningún proceso de reclutamiento, de formación, de gestión ni de control de personal por parte del Jefe de Proyecto. En SAFe los equipos, que se encuentran agrupados en equipos *Scrum* y en Trenes de desarrollo (ART), se autogestionan. Por esta parte, se considera que no es necesario realizar prácticas adicionales para la gestión de los recursos humanos al margen de las que propone SAFe.

En cuanto al resto de recursos del proyecto (materiales, equipamiento, instalaciones, etc.), SAFe no identifica qué rol o en qué momento es necesario llevar a cabo esta tarea. Sin embargo, para asegurar una adecuada gestión de los recursos materiales necesarios para el proyecto, sí se considera necesario contemplar de forma específica esta actividad.

Gestión de las comunicaciones

Uno de los principales objetivos de SAFe es mejorar la coordinación de los equipos en entornos *Agile*. SAFe, sin embargo, en esta área no sugiere artefactos adicionales a los que propone Scrum, los cuales se encuentran basados en la comunicación a través de canales informales (cara a cara y a través de herramientas visuales).

PMBok, por su parte, plantea una gestión formal de las comunicaciones en la que, a partir de una identificación de los interesados, se define qué tipo de información debe recibir cada uno así como el momento y soporte en el que debe recibirla.

La gestión de las comunicaciones que se propone en este trabajo se acerca más a la propia de SAFe/*Agile*. La información debe estar visible y ser accesible por los interesados. Esto permite mejorar la transparencia y la transmisión eficaz de la información que fluye en toda la organización (portfolio, programa, equipo, cliente).

El hecho de trabajar en entornos distribuidos puede parecer un hándicap para que este tipo de comunicación sea eficaz. SAFe, incluso en esos entornos, insiste en que la comunicación directa entre las personas, debe producirse, en especial en el momento de la revisión y la planificación del *PI* (*Product Increment*). Es altamente recomendable que la reunión se convoque de forma centralizada en una ubicación a la que puedan asistir todos los miembros de los equipos (o, al menos, una representación de los mismos). Dependiendo de la ubicación de los equipos y de la frecuencia de la reunión, incluso para entornos distribuidos, esto debería promoverse. Para los casos en los que no sea posible celebrar esa reunión física periódica porque la distancia no lo permite, será necesario establecer la comunicación a través de herramientas colaborativas como video-conferencia, chat o cualquier otro método que facilite la comunicación a distancia.

Una vez definido el alcance del trabajo a realizar, dicha información se incluye en *backlogs* y paneles *kanban* que deben estar visibles y ser accesibles por toda la organización, a través de los que se comprueba el avance del trabajo y las posibles desviaciones sobre el plan previsto.

La propuesta que se hace en este trabajo es mantener este tipo de comunicación al estilo SAFe. En principio puede parecer un riesgo en estos entornos, pero se considera importante educar y potenciar la práctica de este tipo de comunicación abierta que, aporta muchos más beneficios en el largo plazo.

Para garantizar que la información llega a todos los interesados, los roles tipo *Release Train Engineer (RTE)*, *Scrum Master*, *Product Manager*, *Product Owner*, o incluso el *Lean Portfolio Management* (Agile PMO), deben asumir,

como parte de su responsabilidad, el asegurar que, todos los interesados en el proyecto, conocen y tienen acceso a la información disponible. Por este motivo, en esta propuesta se promueve una actitud altamente proactiva de estos roles que permita facilitar esta comunicación.

Gestión de los riesgos

La minimización del riesgo es un valor intrínseco en *Agile*. El hecho de trabajar en iteraciones cortas, la flexibilidad y la transparencia en la que los equipos *Agile* trabajan, hacen que el riesgo de que el producto no se adecue a las necesidades del cliente, se reduzca de forma importante. Además, SAFe, consciente de la importancia de este aspecto, incluye en la agenda de la reunión inicial de *PI (Product Increment) slots* específicos para la identificación de los riesgos y la definición de su posterior tratamiento. Esto permite considerar la gestión de riesgos a un nivel superior al de equipo (nivel de programa).

A pesar de que SAFe sí considera actividades relacionadas con la gestión de riesgos en su proceso, debido a su importancia y a que la responsabilidad del seguimiento y control de los mismos no queda definida, se recomienda reforzar este punto implementando mecanismos semi-formales que permitan el registro, análisis y tratamiento más riguroso de esta actividad.

Gestión de las adquisiciones

Uno de los valores del manifiesto *Agile* es el de dar prioridad a '*la colaboración con el cliente sobre la negociación contractual*'. Este valor entra en conflicto claramente con la gestión formal de la relación con los proveedores que promulga PMI-PMBoK. SAFe, al ser una metodología híbrida, incluye brevemente la gestión con los proveedores y lo hace dentro del marco de las 'Grandes Soluciones'. Sin embargo, SAFe no entra en definir el proceso concreto, como lo hace PMI-PMBoK, sino que simplemente señala el momento en el que es preciso incluir a los proveedores en el proceso de desarrollo y cómo sincronizar su trabajo con los Trenes de desarrollo (ART).

Para el caso objeto de este estudio, se realizarán las siguientes adaptaciones:

- Incorporar la coordinación con los proveedores a 'nivel de Programa'. La gestión de proveedores no solo es necesaria cuando hay varios Trenes de desarrollo trabajado. En muchas ocasiones, se hace necesario coordinar el trabajo de un equipo externo incluso cuando solo existe un equipo de trabajo aislado.
- Incluir mecanismos semi-formales para la gestión de los proveedores. Cuando se trata de relaciones con empresas externas que tienen procedimientos de trabajo diferentes y cuyos servicios van a ser objeto de un coste económico, es preciso regular las relaciones entre ambas compañías, así como detallar el trabajo a realizar (alcance, calidad, entregas, etc.).

Gestión de los interesados

El éxito o fracaso de un proyecto depende del grado de satisfacción de los diferentes interesados en un proyecto. *Agile* y SAFe se centran fundamentalmente en 'el cliente', pero en un proyecto hay muchos más interesados que es preciso atender. Por ejemplo, si un proyecto satisface al cliente, pero los costes se han disparado, el grado de satisfacción de la dirección será escaso. O, por ejemplo, si los costes son reducidos, pero se genera una solución que incrementa la 'deuda técnica', es probable que se genere conflicto con los equipos de desarrollo y/o operaciones.

Dado que SAFe no contempla un proceso formal de identificación de interesados, pero se considera un factor crítico, se propone incorporar a los procedimientos una fase semi-formal de identificación de interesados. En este caso se recomienda que el trabajo sea realizado por perfiles de tipo 'facilitador' como los '*Release Train Engineer (RTE)*'. El RTE realizará una identificación de los interesados y su clasificación en función de su influencia/impacto. Posteriormente, el RTE será el encargado de canalizar las necesidades cada uno a diferentes puntos del proceso (ejemplo: hacia los *Epic Owner* que podrán incluir funcionalidades adicionales a las Épicas, hacia los Arquitectos que incorporarán requisitos adicionales técnicos, etc.).

A modo de resumen, en la siguiente gráfica se incluye una tabla en la que se muestra la comparativa entre los dos *frameworks* (PMI-PMBok y SAFe) y se concretan los puntos que serán reforzados en la hibridación propuesta.

	PMP - PMBoK	AGILE -SAFe	HIBRIDACIÓN PMP-PMBoK / Agile-SAFe
INTEGRACIÓN	Procesos para identificar, definir, combinar, unificar y coordinar las diversas actividades de dirección del proyecto. El jefe de proyecto es el principal responsable de la integración del proyecto.	Actividades dirigidas a la coordinación de todas las partes para lograr 'desarrollar el máximo valor de negocio con el tiempo de entrega sostenible más corto y con el mayor nivel de calidad posible'. La responsabilidad de este proceso recae sobre los roles facilitadores del proceso (scrum master, release train engineer).	Se adoptan los mecanismos de coordinación SAFe.
ALCANCE	Procesos requeridos para garantizar que el proyecto contempla todo el trabajo requerido, y únicamente el trabajo requerido para completar el proyecto con éxito.	Actividades asociadas a la definición de las tareas necesarias para desplegar un producto que satisfaga al cliente. El detalle completo del trabajo requerido no se conoce al comienzo del proyecto sino que se depura a medida que avanzan los desarrollos.	Se adoptan los mecanismos para la definición del alcance de SAFe/Agile.
TIEMPO	Procesos requeridos para asegurar la finalización del proyecto a tiempo.	Incluye los procesos de planificación de los sprints (corto plazo) y Product Increment (medio plazo). Para una visión a más largo plazo se realizan proyecciones a nivel de roadmap.	La gestión del tiempo se realiza en base a los artefactos que define SAFe.
COSTE	Procesos involucrados en estimar, presupuestar, financiar, obtener financiación, gestionar y controlar los costos de modo que se complete el proyecto dentro del presupuesto aprobado. Se emplea la técnica del Valor ganado para monitorizar las desviaciones de coste - tiempo - alcance.	A nivel de portfolio, se incluyen las actividades que permiten seleccionar los flujos de valor a financiar (Lean Business case). A nivel de programa/equipo, se priorizan las actividades en función del WSJF (Weighted Shortest Job First). Es decir, se acometen primero las que aporten más valor con menos esfuerzo.	La gestión del coste se realiza en base a los artefactos SAFe que se refuerzan mediante información adicional que permiten visualizar las desviaciones de alcance-tiempo a nivel de Product Increment.
CALIDAD	Procesos para la planificación, gestión y control de los requisitos de calidad del proyecto y el producto a fin de satisfacer los objetivos de los interesados.	Proceso iterativo de control de calidad (PDCA): la calidad se asegura mediante la ejecución del desarrollo en ciclos cortos de tiempo. También se asegura un balance entre mejoras técnicas y requisitos funcionales en los desarrollos.	La calidad es una de los puntos más fuertes de la metodología SAFe, con lo que no es preciso añadir artefactos adicionales en este sentido.
RECURSOS	Procesos para identificar, adquirir y gestionar los recursos (humanos y materiales) necesarios para la conclusión exitosa del proyecto.	Proceso de asignación de los ART (trenes de desarrollo) a flujos de valor (asignación de RRHH).	En cuanto a los RRHH, no es necesario incorporar herramientas adicionales (los equipos multidisciplinares son el corazón de las filosofías Agile). Se considera necesario documentar las necesidades de recursos materiales del proyecto (identificación de necesidades: Lean Business Case; gestión de los recursos materiales: coordinado por el Release Train Engineer con el resto de áreas implicadas)
COMUNICACIONES	Procesos necesarios para asegurar que las necesidades de información del proyecto y de sus interesados se satisfagan a través del desarrollo de objetos (informes) y de actividades que permitan lograr un intercambio eficaz de información.	Proceso de comunicación directa y visual (radiadores de información)	La gestión de la comunicación en entornos distribuidos, debe ser reforzada con herramientas software de tipo colaborativo. Además, se recomienda reforzar los canales de comunicación: no es suficiente con disponer de 'radiadores de información'. Es preciso que la información llegue a ciertos interlocutores con o que se definen mecanismos de reporte periódicos.
RIESGOS	Procesos para llevar a cabo la identificación, análisis, planificación de respuesta, implementación de respuesta y monitoreo de los riesgos de un proyecto.	Proceso de identificación de riesgos a nivel de PI (Product Increment)	La gestión de riesgos es un aspecto crítico dentro de la gestión de proyectos y se reforzará el proceso con mecanismos semiformales de identificación y tratamiento de los riesgos.
ADQUISICIONES	Procesos necesarios para comprar o adquirir productos, servicios o resultados que es preciso obtener fuera del equipo del proyecto (contratos, órdenes de compra, acuerdos de nivel de servicio (SLAs) internos, etc)	Proceso de coordinación de proveedores a nivel de Solución.	Se adoptan medidas adicionales para reforzar la gestión de los proveedores: - Se incorpora la gestión de los proveedores al nivel de programa. - Se articulan procedimientos semiformales para la gestión de los proveedores.
INTERESADOS	Procesos para la identificación de las personas, grupos u organizaciones que pueden afectar o ser afectados por el proyecto, para analizar las expectativas de los interesados y su impacto en el proyecto, y para desarrollar estrategias de gestión adecuadas a fin de lograr la participación eficaz de los interesados en las decisiones y en la ejecución del proyecto.	SAFe identifica un interesado principal: el cliente. También se puede identificar a 'la compañía' a través de roles como los Business Owners y el Lean Portfolio Management. SAFe no incluye mecanismos adicionales para la identificación de interesados adicionales.	Se refuerza el procedimiento de identificación de interesados con mecanismos semiformales que serán llevados a cabo por el Release Train Engineer.

Tabla 3. Comparativa PMI-PMBoK vs SAFe

4.4. Procedimiento de gestión de proyectos

A continuación, se presentan los procedimientos de gestión de proyectos que han sido desarrollados con aportaciones de SAFe (*Agile*) y PMI-PMBok (predictiva). El trabajo se ha realizado tomando como base la comparativa incluida en el apartado 4.3. *Hibridación PMI-PMBok / SAFe* de este documento.

Los procedimientos desarrollados son los siguientes:

- Procedimiento de gestión del portfolio
- Procedimiento de gestión de proyectos (incluye el nivel de programa y de equipo)

En este trabajo no se contempla la redacción formal de los procedimientos, cuya elaboración sería objeto de un trabajo en sí mismo. En lugar de dicha redacción formal, los procedimientos se presentan mediante una breve descripción introductoria y un esquema en el que se concretan sus componentes fundamentales (fases que lo forman, técnicas y herramientas recomendadas, etc.). Además, también se incluye el diagrama de flujo que representa la sucesión de las actividades principales del proceso y los puntos de decisión principales.

Procedimiento de gestión del portfolio

En este procedimiento se definen las actividades relacionadas con la selección y priorización de las iniciativas que serán acometidas en la compañía. Este es un proceso muy cercano a la dirección estratégica y es un nivel donde se exige un mayor grado de predictibilidad y control (plazos, coste, etc.). El procedimiento desarrollado, mantiene una filosofía predominante de tipo *Agile*. Sin embargo, las expectativas de la dirección hacen imprescindible encontrar un equilibrio entre las dos perspectivas (*Agile* vs Predictibilidad). Hay que tener en cuenta que, para alcanzar ese punto medio, es preciso una mayor flexibilidad de la dirección en ese sentido.

Para el caso de las metodologías tradicionales, la gestión del portfolio es responsabilidad de la PMO (*Project Management Office*) que, teniendo en cuenta las directrices estratégicas de la compañía, centraliza actividades como la definición de requisitos, la planificación de los proyectos y el reporte de los resultados. En muchas empresas que usan metodologías *Agile*, esta tarea se

encuentra descentralizada de forma intencionada con el objetivo de distribuir la toma de decisiones entre los diferentes actores del proceso. Sin embargo, otras empresas optan por la creación de una APMO (PMO *Agile*) que, junto con los Jefes de proyecto, sean los que realicen este rol operativo a nivel de portfolio.

Para la efectiva gestión del portfolio es necesario además definir unos mecanismos de diseño de *roadmap* y financiación ligeros (*Lean*) que permitan cambiar y ajustar los planes de manera flexible.

Procedimiento de gestión de programa/proyecto

Una vez que las distintas iniciativas planteadas dentro de la compañía han sido aprobadas, analizadas y convenientemente priorizadas, quedan a la espera para ser desarrolladas. El procedimiento de gestión de programa/proyecto comienza en el momento en el que hay recursos disponibles para ejecutar el trabajo y tiene el objetivo de sincronizar a todos los equipos que participan en el desarrollo para lograr los objetivos planteados. Los procedimientos, a este nivel, son básicamente *Agile*: los equipos trabajan de forma coordinada y en base a iteraciones, cuyo resultado se despliega a un ritmo constante.

Uno de los roles principales en este nivel es el '*Release Train Engineer*'. Sus funciones son las de facilitar la reunión de *PI (Product Increment)* y coordinar a todos los equipos que están trabajando. Además, también gestionan los riesgos y las dependencias e intentan guiar el proceso para obtener un aceptable grado de predictibilidad. Esa labor es muy similar a la realizada por un Jefe de Proyecto 'tradicional', con lo que resulta adecuado hacer esta equivalencia entre ambas metodologías: '*Release Train Engineer* ↔ *Jefe de Proyecto*'. Para que un Jefe de Proyecto tradicional esté preparado para realizar las tareas que SAFe asigna al *Release Train Engineer*, es preciso una adecuada formación, sobre todo en aspectos relacionados con 'Gestión de equipos y técnicas de facilitación'. Este rol dentro de la metodología se denominará '*Agile Project Manager (APM)*'. Además, también será el encargado de realizar la gestión de riesgos y de comunicar el avance de los desarrollos al nivel de portfolio.

GESTIÓN DEL PORTFOLIO	
ALCANCE	
En este procedimiento se definen las distintas actividades relacionadas con la selección y priorización de las actuaciones o proyectos que serán acometidos en la compañía permitiendo, de esa forma, alinear los mecanismos de producción software con la estrategia de negocio.	
DEFINICIONES	
<p><u>Épica-portfolio</u>: Descripción de una iniciativa de alto nivel, alineada con la estrategia de la compañía.</p> <p><u>Lean Business Case (LBC)</u>: Business Case sencillo que incluye información tanto funcional como financiera: se definen los objetivos de la Épica, los resultados que se prevé obtener, los costes y plazos estimados para el desarrollo, etc.</p> <p><u>MVP</u> (Minimum Viable Product): Descripción del mínimo producto que podría ponerse a disposición del cliente.</p> <p><u>Comité evaluación portfolio (CEP)</u>: Equipo de personas con funciones de dirección en la compañía que se reúnen periódicamente para evaluar las nuevas iniciativas a acometer, priorizarlas y evaluar las que están en curso. Este comité realiza tareas asociadas a la función 'Lean Portfolio Management'.</p>	
RESPONSABILIDADES	
<p><u>APMO (Agile Project Management Office)</u>: Registrar y presentar las nuevas iniciativas al Comité de Evaluación del Portfolio (CEP). Elaboración de informes que permitan dar seguimiento a las iniciativas en marcha.</p> <p><u>Miembros del CEP</u>: Acudir a las reuniones del comité o nombrar a sustitutos con poder de toma de decisiones. Evaluar las iniciativas presentadas para su aprobación y seguimiento de las iniciativas en curso.</p> <p><u>Roles de SAFe a nivel de portfolio/programa</u>: Epic Owner, Arquitectos (empresa y sistema), Product manager, Business Owner)</p>	
FASES	HERRAMIENTAS Y TÉCNICAS
<p>Fase de iniciación</p> <ul style="list-style-type: none"> - Identificar iniciativas de interés para la compañía (Responsables: Product Manager) - Priorizar las iniciativas y estimarlas a alto nivel (WIP máximo) (Responsable: Epic Owners, product managers, arquitectos, etc) - Seleccionar las iniciativas prioritarias para realizar un análisis (Responsable: APMO, Product Manager) 	<p>Tormenta de ideas</p> <p>User Persona (arquetipo)</p> <p>Mapa de empatía</p> <p>Modelo de negocio Canvas</p> <p>MoSCoW</p>
<p>Fase de análisis</p> <ul style="list-style-type: none"> - Realizar un análisis más detallado de la iniciativa (Responsable: Epic Owner) <ul style="list-style-type: none"> > Definir el alcance de manera sencilla en forma de épica que incluya la formulación de una hipótesis (lo que se espera conseguir) > Elaborar un breve Business Case (Lean Business Case - LBC) incluyendo la identificación de posibles adquisiciones y/u otros recursos necesarios. > Realizar una identificación inicial de los riesgos de la Épica. > Definir el posible MVP (Minimum Viable Product). > Presentar la épica a aprobación del CEP (Responsable: APMO) 	<p>Lean Business Case</p> <p>Impact mapping</p> <p>Coste del retraso</p> <p>Análisis valor vs complejidad</p>
<p>Fase de revisión/exploración (tras la aprobación de la iniciativa)</p> <ul style="list-style-type: none"> - Incorporar la épica al Backlog de portfolio (Responsable: Epic Owner) - Seleccionar el/los ART (trenes de desarrollo) que desarrollarán la solución (Responsable: CEP-Lean Portfolio Management). - Priorizar el Backlog en base al WSJF u otro criterio (Responsable: CEP-Lean Portfolio Management) - Depurar y detallar con mayor precisión el MVP (Responsable: Epic Owner) 	<p>WSJF</p> <p>MoSCoW</p> <p>Otras técnicas de priorización</p>
<p>Fase de desarrollo</p> <ul style="list-style-type: none"> - Trasladar las épicas al backlog de programa y comenzar con el proceso de desarrollo software (articulado según el procedimiento 'Gestión de Programa') - Reportar el avance de la ejecución de los desarrollos a medida que la ejecución del proceso de Gestión de programa va avanzando. 	<p>Diagrama de flujo acumulado</p> <p>Diagrama Burn-up de Coste-Valor</p> <p>Informe de progreso de las épicas</p> <p>Valor ganado Agile</p>
<p>Fase de cierre</p> <ul style="list-style-type: none"> - Analizar los resultados de las épicas y cerrar el proceso. 	<p>Visualización y análisis de datos (cualitativo, cuantitativo)</p>
REGISTROS	
<ul style="list-style-type: none"> - Backlog de portfolio (descripción de la épica, hipótesis de negocio, riesgos, MVP) - Kanban Portfolio (flujo del proceso) - Lean Business Case - Informe de cierre (revisión del Lean Business Case incluyendo datos de los objetivos alcanzados y los resultados obtenidos) 	

Tabla 4. Procedimiento híbrido para la gestión del portfolio.

GESTIÓN DEL PROGRAMA	
ALCANCE	
En este procedimiento se definen las distintas actividades relacionadas con el desarrollo y despliegue del incremento de valor continuo en la compañía. El objetivo es conseguir la coordinación y sincronización de todos los equipos para lograr completar dicha entrega.	
DEFINICIONES	
<u>Feature/Enabler</u> : Historias en las que se dividen las épicas de portfolio y que forman parte del backlog de programa. <u>QA (Quality assurance)</u> : Actividades de evaluación del desarrollo software para garantizar que el producto final sea de calidad.	
RESPONSABILIDADES	
<u>APMs (Agile Project Managers)-Release Train Manager</u> : En colaboración con los Product Managers, realizar la función coordinadora a nivel de PI, gestión de riesgos, facilitación de reuniones, reporte del avance del trabajo. <u>Roles de SAFe a nivel de programa/equipo</u> : Arquitectos (empresa y sistema), Product manager, Business Owner, equipos Scrum.	
FASES	HERRAMIENTAS Y TÉCNICAS
Fase de definición <ul style="list-style-type: none"> - Realizar la identificación de interesados y de riesgos (Responsable: APM) - Revisar el contenido del backlog de portfolio, priorizarlo y prepararlo para la celebración de la reunión de PI (Product Increment) (Responsable: APM, Product Manager, Epic Owner) - Establecer los objetivos de negocio para el PI (Responsable: Business Owner) - Identificar necesidades y/o dependencias derivadas de los objetivos del PI, controlarlas y gestionarlas (relación con proveedores, compra de material, redacción de contratos) (Responsable: APM) - Preparar y convocar la reunión de PI (ART y resto de involucrados) (Responsable: APM) - Asegurar que la reunión de PI es efectiva y produce los resultados esperados: Backlog de de programa con alcance del PI, MVP, dependencias, revisión y gestión de riesgos por los equipos. (Responsable: APM) - Obtener el compromiso por parte del equipo con los objetivos del PI acordados. 	Análisis de interesados WSJF Tormenta de ideas, juicio de expertos, votaciones, etc Categorización de riesgos, análisis de probabilidad-impacto, respuesta a los riesgos Técnicas negociación, análisis de propuestas, etc
Fase de implementación <ul style="list-style-type: none"> - Trasladar los objetivos del PI a los diferentes equipos Scrum y resto de personas implicadas en los desarrollos (Responsable: APM, Product Owners) - Coordinar las actividades de los distintos equipos durante la ejecución del PI (Responsable: APM, Product Owners, Scrum masters) - Revisar el contenido del Backlog de programa para ajustar, si fuera necesario, las prioridades de cada sprint (Responsable: Product Owners) - Desarrollar el alcance del PI (Responsable: Equipos Scrum) - Informar sobre los resultados parciales del PI (transparencia con el trabajo realizado, identificación temprana de problemas, etc) (Responsable: APM) 	Diagrama de flujo acumulado Diagrama Burn-up de Coste-Valor Informe de progreso de las épicas Valor ganado Agile
Fase de pruebas, revisión, ajustes y despliegue (Responsable: Equipos scrum) <ul style="list-style-type: none"> - Realizar la fase de QA (Quality assurance) de los desarrollos para asegurar la calidad. - Integrar, al menos una vez al finalizar cada iteración, el contenido desarrollado y convenientemente testeado. - Celebrar reuniones demo a nivel de iteración para validar los resultados parciales del PI. - Celebrar las reuniones demo al finalizar el PI (Product Increment) para validar los resultados totales del PI. Ejecución de pruebas de aceptación, si fuera necesario. - Desplegar los desarrollos de forma continua (al menos una vez al finalizar la iteración) 	Herramientas para facilitar la ejecución de pruebas de software (unitarias, stress, integración, funcionales) Herramientas y técnicas de soporte a CI/CD (Continuous Integration/Continuous Deployment)
Fase de de entrega <ul style="list-style-type: none"> - Desplegar los desarrollos en producción según la estrategia definida (cada PI, con más o menos frecuencia o bajo demanda) 	Herramientas y técnicas de soporte a CD (Continuous Deployment)
REGISTROS	
<ul style="list-style-type: none"> - Tablero de programa - Backlog de equipos - Kanban de equipos (flujo del proceso) - Identificación de interesados - Identificación y seguimiento de riesgos - Contratos con proveedores - Registro de requisitos y suministro de recursos materiales 	

Tabla 5. Procedimiento híbrido para la gestión de programa-proyecto.

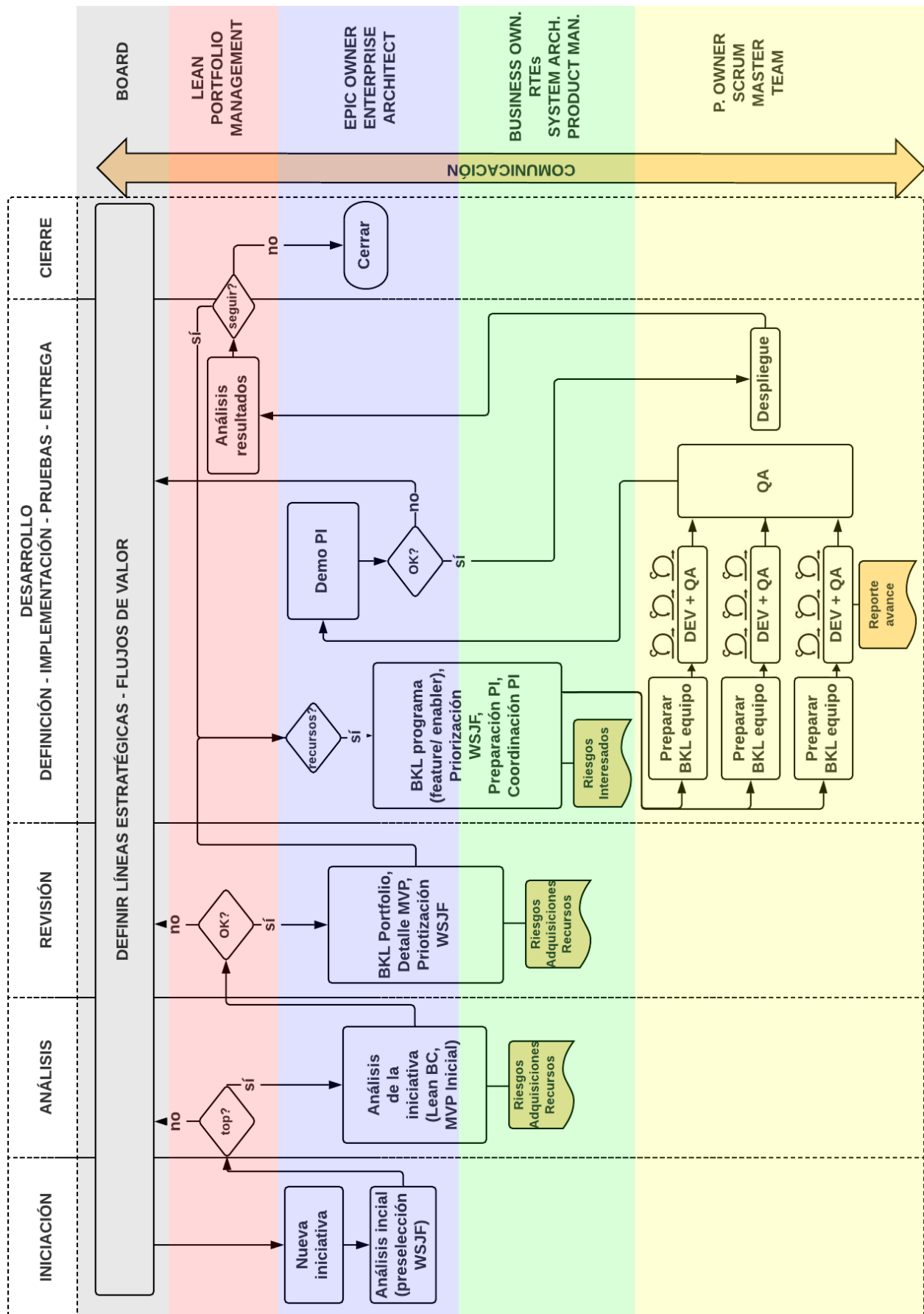


Figura 22. Diagrama de flujo de los procedimientos híbridos.

4.5. Implantación del procedimiento

Para finalizar el trabajo, se presenta un escenario donde podría aplicarse este procedimiento. También se incluyen recomendaciones para guiar un proceso de implantación satisfactorio.

El trabajo desarrollado está orientado a empresas de tamaño medio-grande del sector TIC que dispongan de equipos de desarrollo multidisciplinares con experiencia en distintos ámbitos (desarrollo software, operaciones, redes y comunicaciones, etc.). Los equipos de desarrollo pueden trabajar en el mismo centro de trabajo o en ubicaciones distribuidas. Además, también se plantea la posible subcontratación de proveedores o de personal temporal que soporte ciertas actividades. El número de empleados y/o colaboradores encargados de desarrollar las soluciones es, como mínimo, de 50 personas (número mínimo recomendado para poder formar un ART). Para empresas de menor tamaño, sería recomendable hacer una simplificación de los procedimientos ya que los actuales podrían resultar demasiado complejos.

Además, la empresa debe tener madurez en temas relacionados con gestión de proyectos, en concreto, en metodologías de carácter predictivo. También resulta conveniente que los equipos de desarrollo tengan cierta experiencia en metodologías *Agile* y lleven un tiempo experimentando con dichas prácticas y con las limitaciones que presenta cuando se aplica en entornos complejos.

Este escenario, a pesar de que parece muy concreto, resulta el camino habitual que han recorrido muchas empresas del sector: comenzando con una gestión de proyectos tradicional y evolucionado hacia la adopción de prácticas *Agile* a nivel operativo con el objetivo de buscar rapidez de respuesta a las necesidades del mercado. Durante ese camino, se han enfrentado a la mayoría de los problemas expuestos anteriormente.

Las empresas que se encuentren en esa situación y deseen conseguir una mayor eficiencia en sus procesos, deberán abordar un proceso de implantación de *Agile* a nivel corporativo, lo cual incluye la adopción de prácticas de escalado *Agile*. SAFe es un marco muy completo y también incluye dentro de su extensa documentación recomendaciones para una exitosa implantación del *framework*. En la siguiente figura se representa el *roadmap* propuesto por SAFe para la implantación del proceso.

SAFe® Implementation Roadmap

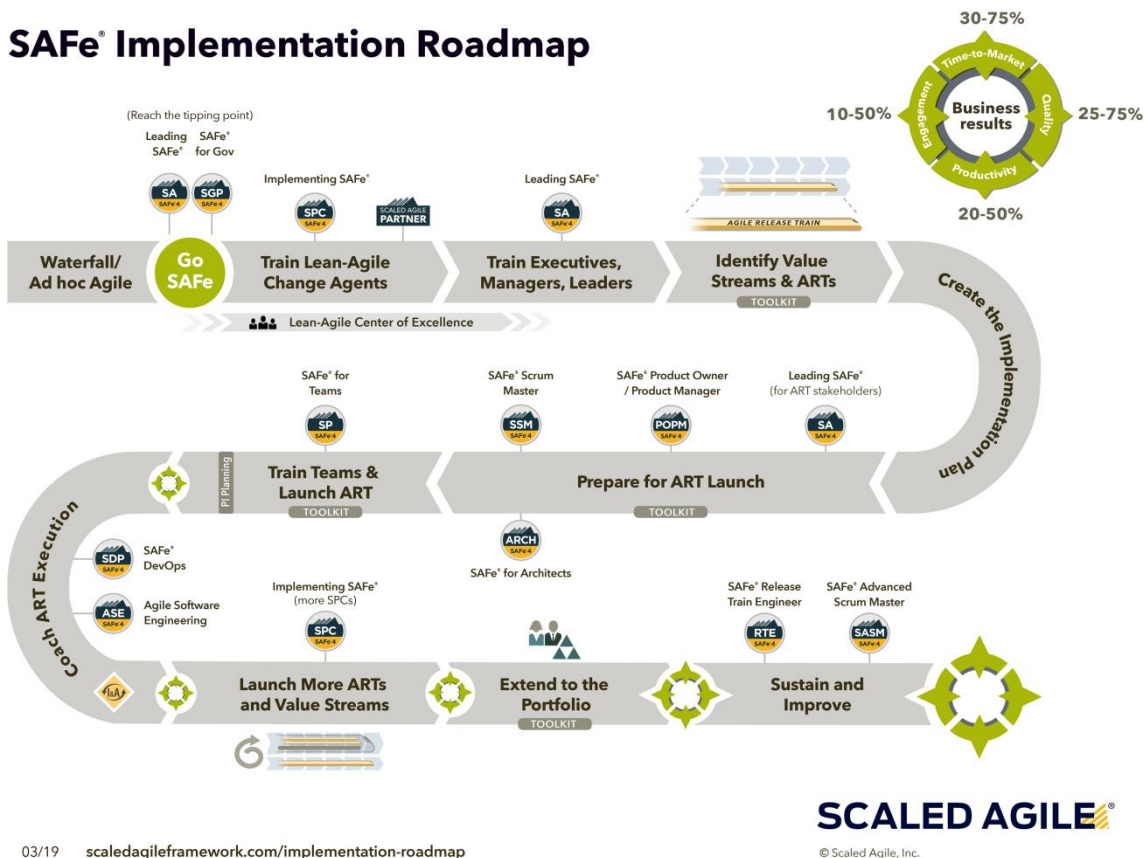


Figura 23. Roadmap para la implementación de SAFe (23)

SAFe, como puede observarse, recomienda realizar una implantación gradual, comenzando con el nivel de equipo para extender posteriormente su aplicación a niveles superiores (programa/portfolio). Además, a pesar de que la implantación a nivel de portfolio se hace en fases avanzadas del proceso, tanto la formación de los ejecutivos y mandos de la compañía como la identificación de los *flujos de valor/Value Streams*, se realiza en las fases iniciales del proceso. En ese aspecto, la propuesta realizada en este trabajo coincide en general con la propuesta de SAFe.

Teniendo en cuenta la experiencia propia y la información obtenida del framework SAFe, se realizan las siguientes recomendaciones para el proceso de implantación de los procedimientos desarrollados:

- El proceso de cambio debe partir de una necesidad de mejora considerando que el cambio por el cambio no funciona.

- El proceso de cambio debe ser promovido desde la dirección de la empresa. La dirección debe estar convencida de los beneficios que se obtendrán tras el proceso y debe apoyarlo.
- Todas las áreas de la empresa deben estar alineadas en la misma dirección y deben apoyar las decisiones adoptadas por la gestión de la empresa.
- Debe elaborarse un plan de implantación liderado por un equipo de personas con distintos perfiles y con credibilidad dentro de la compañía. En ocasiones es importante contar también con personal externo que aporte recomendaciones desde un punto de vista neutral y objetivo. En este plan deben identificarse métricas que permitan seguir la evolución del proceso y sus resultados.
- Los equipos deben tener experiencia suficiente en prácticas *Agile* (incluyendo prácticas de DevOps que faciliten la integración y el despliegue continuo)
- El realizar una adecuada gestión de las expectativas es fundamental para acometer el cambio con garantías de éxito. Las distintas partes involucradas en el cambio deben conocer y aceptar la transformación que va a acometerse y sus implicaciones.

Teniendo en cuenta todo lo anterior, se propone la siguiente planificación para la implantación del proceso completo que se plantea a lo largo de tres años.

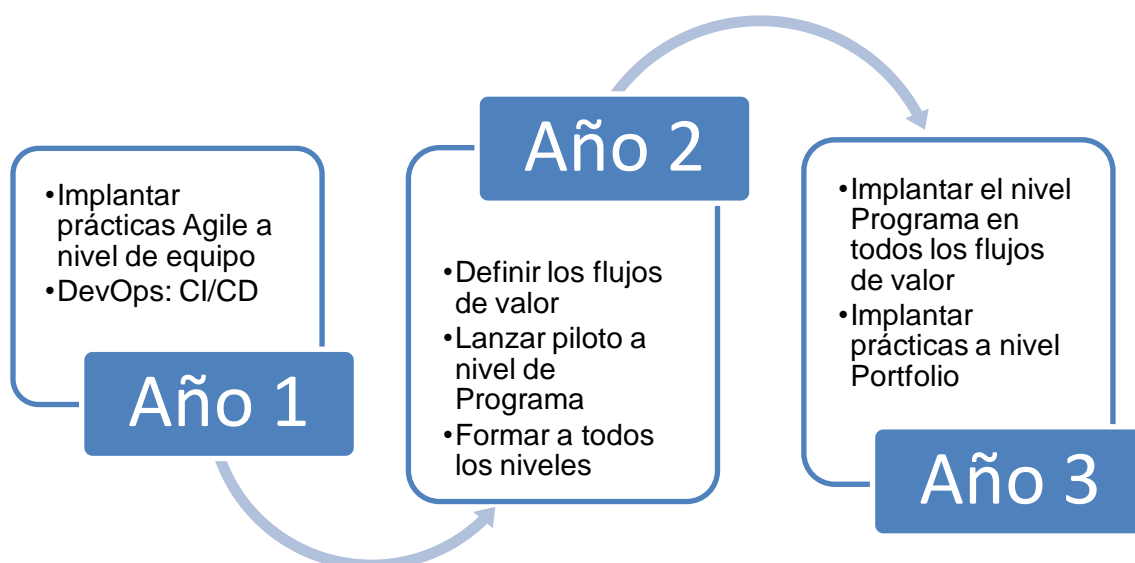


Figura 23. Línea de tiempo para una transformación Agile

5. CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS

La rápida velocidad con la que avanza la tecnología hoy día tiene repercusiones importantes en la sociedad en la que vivimos. En el mundo empresarial, las compañías que quieren sobrevivir se encuentran ante la exigencia de responder de forma rápida a dichos cambios. Aunque son las empresas del sector TIC las más impactadas por este fenómeno, es difícil encontrar un sector que no se vea directamente influenciado por la revolución de las tecnologías de la información y las telecomunicaciones.

En este entorno, los procedimientos tradicionales que antes funcionaban, ya no resultan eficaces. Las empresas no pueden dedicar años de trabajo a madurar productos que estarían obsoletos antes de salir al mercado. Como respuesta a esta necesidad de adaptación al cambio, surgieron las metodologías *Agile*. En principio, aunque estas metodologías se orientaban al desarrollo software, su filosofía se ha estado extendiendo a muchos otros campos del conocimiento en los cuales están surgiendo (o tomando fuerza) prácticas de similar orientación (p. ej. design thinking).

La aplicación de estas técnicas no es sencilla y la principal dificultad no radica en la adopción de nuevos procedimientos de trabajo, sino en el cambio cultural que requiere. Este cambio cultural se supone más complicado cuantas más personas o niveles involucra. Por ejemplo, para empresas pequeñas, o en el ámbito de pequeños equipos de trabajo, es más fácil conseguir la implicación y confianza que exige la adopción de *Agile*. Sin embargo, en grandes corporaciones altamente burocratizadas en las que existen intereses contrapuestos y silos funcionales, poner en marcha esta filosofía resulta muy complicado.

La información presentada en este trabajo permite confirmar la anterior hipótesis. Según los estudios analizados, ¿Cuál es la razón de la dificultad de implantar *Agile* a gran escala? Los problemas que se señalan son los siguientes: una falta de comunicación, una gestión de las dependencias insuficiente y la posible deficiencia de las soluciones técnicas adoptadas. En realidad, el problema de fondo que trasciende de los anteriores motivos, es una escasa colaboración entre los equipos de trabajo, bien sea por la dificultad de las relaciones en grandes equipos que se encuentran, en ocasiones, en

distintas localizaciones, o bien por limitaciones internas derivadas de la cultura de la compañía.

Los métodos de escalado *Agile* plantean soluciones a los problemas anteriores introduciendo mecanismos que permiten alinear a los equipos de trabajo. El objetivo es orientar todos los esfuerzos en la misma dirección, fomentado, asimismo, la colaboración para mejorar la agilidad en los procesos. A través de diferentes roles y artefactos, los métodos de escalado *Agile* dirigen las relaciones en estos entornos más grandes, facilitando, de esa forma, la adopción de estas metodologías.

El proceso de implantación de métodos de escalado *Agile* requiere de un tiempo de experimentación, tras el cual, es recomendable realizar adaptaciones en función de las necesidades propias. En este trabajo, partiendo de un estudio detallado de SAFe, como marco de trabajo de escalado *Agile*, y de PMI-PMBok, como metodología con alto control sobre los procesos de gestión de proyectos, se propone una *tailorización* de los procedimientos concretando algún aspecto de los procesos de gestión de costes, recursos, comunicaciones, riesgos, adquisiciones e interesados.

Esta propuesta se realiza, no solo en base al estudio teórico descrito, sino también a la experiencia propia adquirida en la implantación de prácticas *Agile* en entornos complejos de trabajo. Aunque es cierto que cada caso es diferente y que cada compañía debería experimentar y descubrir qué prácticas necesita adoptar o adaptar, las recomendaciones realizadas en este trabajo son muy genéricas y podrían ser de utilidad para empresas con amplia experiencia en entornos predictivos, con cierta andadura en prácticas *Agile* y con necesidad de adoptar prácticas de escalado *Agile*. Por lo tanto, como futura línea de trabajo derivada de este estudio, se podría plantear la puesta en marcha de estos procedimientos con el objetivo de testarlos y validarlos en un entorno real.

Otra posible línea de investigación interesante que podría derivar de este trabajo, es la realización de un estudio en el que se plantee la compatibilidad de estos procedimientos (en general, de los procedimientos tipo *Agile*) con normas y certificaciones ISO. Actualmente no hay ninguna norma ISO que contemple las metodologías tipo *Agile* (la norma ISO21500 de gestión de proyectos, sigue las recomendaciones de PMBoK). Sin embargo, las certificaciones de calidad y seguridad, como la ISO9001 e ISO27001, son de

aplicación frecuente en el sector TIC y, es en este escenario, en el que se considera interesante estudiar su compatibilidad. En realidad, las metodologías *Agile* y las normas ISO no deberían ser incompatibles, ya que ambas tienen objetivos distintos. Mientras las normas ISO definen el ‘*qué*’ debe encontrarse en los procesos, las metodologías establecen el ‘*cómo*’ hacerlo. Sin embargo, en las metodologías de tipo *Agile*, donde se minimiza la documentación y ciertos procesos se basan en soportes volátiles (paneles kanban físicos), puede resultar complicado disponer de registros de que los procesos se han ejecutado (el ‘*qué*’ se ha llevado a cabo).

Las filosofías *Agile* están en auge. Cada vez son más compañías y sectores los que se apuntan a esta tendencia. El escenario que plantean es muy atractivo: grupos de trabajo altamente motivados, alineados estratégicamente con la empresa y que trabajan de forma auto-gestionada, se encargan de desarrollar y desplegar, de forma rápida y continua, productos valiosos para el cliente. El camino que hay que recorrer hasta alcanzar ese objetivo no es fácil. Es necesario tiempo, plantear una adaptación gradual y lograr una gran madurez de toda la organización (no solo de los directivos, sino también de los empleados que deben gestionar su autonomía de forma adecuada).

La necesidad de rapidez y adaptación al cambio es una realidad hoy día y, en vista de las previsiones futuras, las empresas que no sean capaces de seguir el ritmo marcado, desaparecerán. Charles Darwin afirmaba en su libro ‘*La teoría de las especies*’: “*Las especies que sobreviven no son las más fuertes, ni las más rápidas, ni las más inteligentes; sino aquellas que se adaptan mejor al cambio*”. Podemos decir que esta cita, aplicada al campo empresarial, es válida. Las empresas que no solo se adapten al cambio, sino que, además, aporten inteligencia (o creatividad) y rapidez, tendrán, además, altas probabilidades de éxito (no solo de supervivencia).

6. BIBLIOGRAFÍA

1. Project Management Institute. PMBOK® Guide. Pennsylvania: Project Management Institute; 2017. 756 p.
2. chaos_report_1994.pdf [Internet]. [citado 6 de julio de 2019]. Disponible en:
https://www.standishgroup.com/sample_research_files/chaos_report_1994.pdf
3. History: The Agile Manifesto [Internet]. [citado 6 de julio de 2019]. Disponible en: <http://agilemanifesto.org/history.html>
4. Serrador P, Pinto JK. Does Agile work? — A quantitative analysis of agile project success. International Journal of Project Management. 1 de julio de 2015;33(5):1040-51.
5. Campanelli AS, Bassi D, Parreiras FS. Agile Transformation Success Factors: A Practitioner's Survey. En: Dubois E, Pohl K, editores. Advanced Information Systems Engineering. Springer International Publishing; 2017. p. 364-79. (Lecture Notes in Computer Science).
6. Bass JM. How product owner teams scale agile methods to large distributed enterprises. Empir Software Eng. 1 de diciembre de 2015;20(6):1525-57.
7. Sekitoleko N, Evbota F, Knauss E, Sandberg A, Chaudron M, Olsson HH. Technical Dependency Challenges in Large-Scale Agile Software Development. En: Cantone G, Marchesi M, editores. Agile Processes in Software Engineering and Extreme Programming. Springer International Publishing; 2014. p. 46-61. (Lecture Notes in Business Information Processing).
8. Dingsøyr T, Moe NB. Towards Principles of Large-Scale Agile Development. En: Dingsøyr T, Moe NB, Tonelli R, Counsell S, Gencel C, Petersen K, editores. Agile Methods Large-Scale Development, Refactoring, Testing, and Estimation. Springer International Publishing; 2014. p. 1-8. (Lecture Notes in Business Information Processing).
9. Griffiths M. Pmi Acp Exam Prep. Edición: 2. Minnetonka, MN: Rmc Pubns Inc; 2015. 434 p.

10. The New Methodology [Internet]. martinowler.com. [citado 6 de julio de 2019]. Disponible en: <https://martinowler.com/articles/newMethodology.html>
11. 2017-Scrum-Guide-US.pdf [Internet]. [citado 6 de julio de 2019]. Disponible en: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100>
12. Blankenship J, Bussa M, Millett S. eXtreme Programming. En: Blankenship J, Bussa M, Millett S, editores. Pro Agile NET Development with Scrum [Internet]. Berkeley, CA: Apress; 2011 [citado 6 de julio de 2019]. p. 29-51. Disponible en: https://doi.org/10.1007/978-1-4302-3534-7_3
13. sm_proyecto.pdf [Internet]. [citado 6 de julio de 2019]. Disponible en: https://www.scrummanager.net/files/sm_proyecto.pdf
14. Špundak M. Mixed Agile/Traditional Project Management Methodology – Reality or Illusion? Procedia - Social and Behavioral Sciences. 19 de marzo de 2014;119:939-48.
15. Gill AQ, Henderson-Sellers B, Niazi M. Scaling for agility: A reference model for hybrid traditional-agile software development methodologies. Inf Syst Front. 1 de abril de 2018;20(2):315-41.
16. Heikkilä VT, Paasivaara M, Lasssenius C, Damian D, Engblom C. Managing the requirements flow from strategy to release in large-scale agile development: a case study at Ericsson. Empir Software Eng. 1 de diciembre de 2017;22(6):2892-936.
17. Evbota F, Knauss E, Sandberg A. Scaling up the Planning Game: Collaboration Challenges in Large-Scale Agile Product Development. En: Sharp H, Hall T, editores. Agile Processes, in Software Engineering, and Extreme Programming. Springer International Publishing; 2016. p. 28-38. (Lecture Notes in Business Information Processing).
18. Papadopoulos G. Moving from Traditional to Agile Software Development Methodologies Also on Large, Distributed Projects. Procedia - Social and Behavioral Sciences. 12 de febrero de 2015;175:455-63.
19. Theocharis G, Kuhrmann M, Münch J, Diebold P. Is Water-Scrum-Fall Reality? On the Use of Agile and Traditional Development Practices. En: Abrahamsson P, Corral L, Oivo M, Russo B, editores. Product-Focused

- Software Process Improvement. Springer International Publishing; 2015. p. 149-66. (Lecture Notes in Computer Science).
20. Project Management Institute, Agile Alliance. Agile Practice Guide. Pennsylvania: Project Management Institute; 2017. 167 p.
 21. Putta A, Paasivaara M, Lassenius C. Benefits and Challenges of Adopting the Scaled Agile Framework (SAFe): Preliminary Results from a Multivocal Literature Review. En: Kuhrmann M, Schneider K, Pfahl D, Amasaki S, Ciolkowski M, Hebig R, et al., editores. Product-Focused Software Process Improvement. Springer International Publishing; 2018. p. 334-51. (Lecture Notes in Computer Science).
 22. 13th Annual State Of Agile Report [Internet]. [citado 6 de julio de 2019]. Disponible en: <https://explore.versionone.com/state-of-agile/13th-annual-state-of-agile-report>
 23. Scrum of Scrums [Internet]. Agile Alliance. 2015 [citado 6 de julio de 2019]. Disponible en: <https://www.agilealliance.org/glossary/scrum-of-scrums/>
 24. Large Scale Scrum (LeSS) [Internet]. [citado 6 de julio de 2019]. Disponible en: <https://less.works/>
 25. EOI_LeanManufacturing_2013.pdf [Internet]. [citado 6 de julio de 2019]. Disponible en: https://api.eoi.es/api_v1_dev.php/fedora/asset/eoi:80094/EOI_LeanManufacturing_2013.pdf
 26. Disciplined Agile (DA) [Internet]. Disciplined Agile (DA). [citado 6 de julio de 2019]. Disponible en: <http://disciplinedagiledelivery.com/>
 27. Kniberg H, Ivarsson A. with Tribes, Squads, Chapters & Guilds. octubre de 2012;14.
 28. Knaster R, Leffingwell D. SAFe 4.5 Distilled: Applying the Scaled Agile Framework for Lean Enterprises. Edición: new edition. Boston, MA: Addison-Wesley Educational Publishers Inc; 2018. 352 p.
 29. Scaled Agile Framework – SAFe for Lean Enterprises [Internet]. [citado 6 de julio de 2019]. Disponible en: <https://www.scaledagileframework.com>
 30. Alqudah M, Razali R. A Review of Scaling Agile Methods in Large Software Development. International Journal on Advanced Science, Engineering and Information Technology. 25 de diciembre de 2016;6:828.